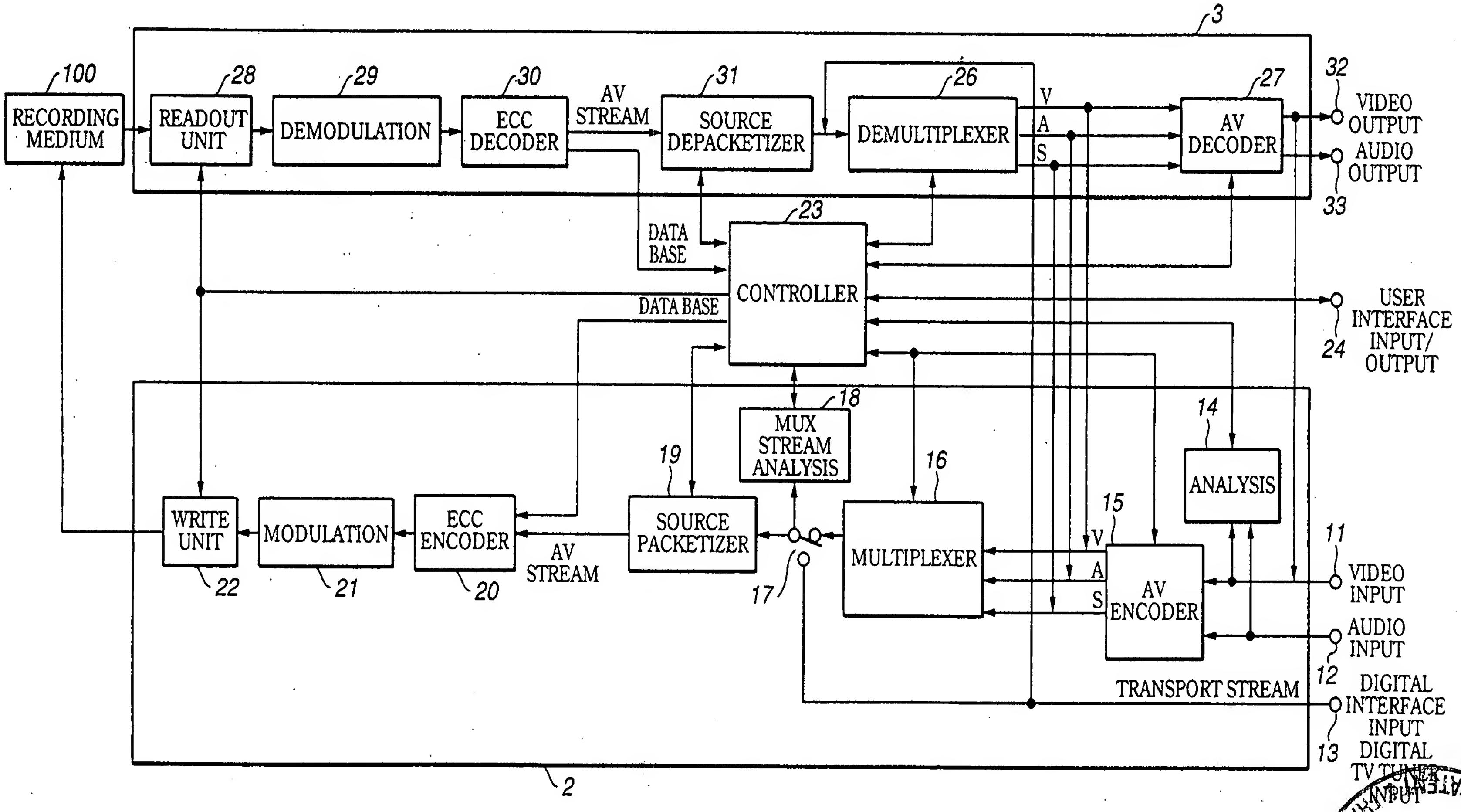

100
RECORDING MEDIUM
28
READOUT UNIT
29
DEMODULATION
30
ECC DECODER
AV STREAM
31
SOURCE DEPACKETIZER
26
DEMULTIPLEXER
V
A
S
27
AV DECODER
3
32
VIDEO OUTPUT
AUDIO OUTPUT
33
DATA BASE
23
CONTROLLER
DATA BASE
USER INTERFACE INPUT/ OUTPUT
24
18
MUX STREAM ANALYSIS
19
SOURCE PACKETIZER
16
MULTIPLEXER
15
V
A
S
AV ENCODER
14
ANALYSIS
11
VIDEO INPUT
AUDIO INPUT
12
TRANSPORT STREAM
DIGITAL INTERFACE INPUT
13
DIGITAL TV TUNER INPUT
WRITE UNIT
22
MODULATION
21
ECC ENCODER
20
AV STREAM
17
2

# FIG.1

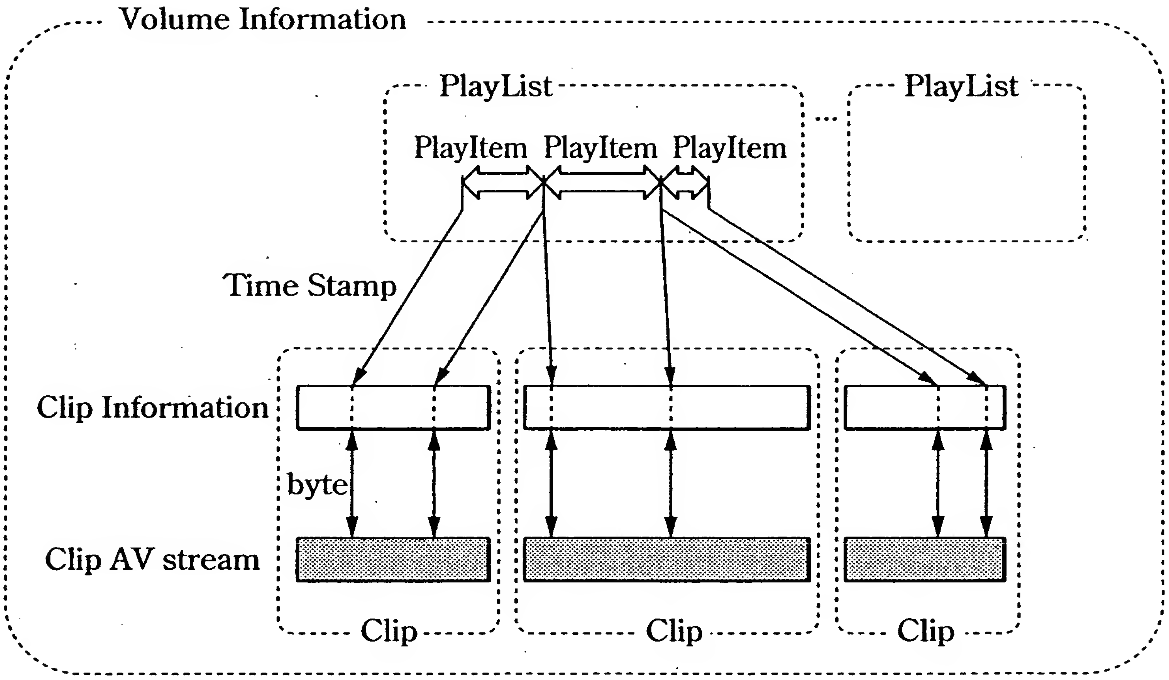

Volume Information
PlayList
PlayItem
PlayItem
PlayItem
...
PlayList
Time Stamp
Clip Information
byte
Clip AV stream
Clip
Clip
Clip

FIG.2

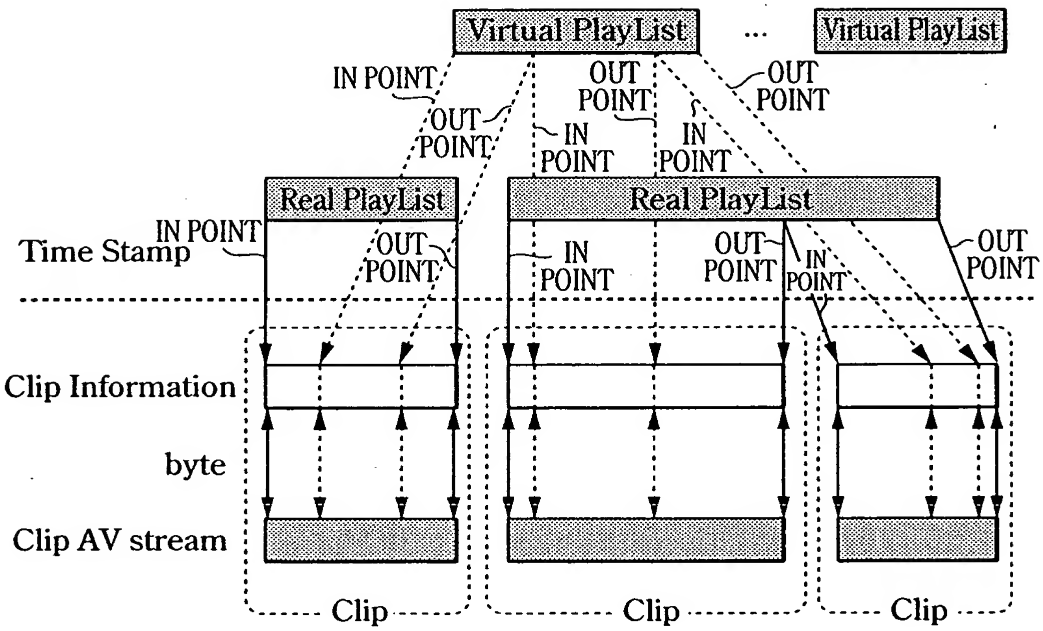

Virtual PlayList
...
Virtual PlayList
IN POINT
OUT POINT
IN POINT
OUT POINT
IN POINT
OUT POINT
Real PlayList
Real PlayList
Time Stamp
IN POINT
OUT POINT
IN POINT
OUT POINT
IN POINT
OUT POINT
Clip Information
byte
Clip AV stream
Clip
Clip
Clip

FIG.3

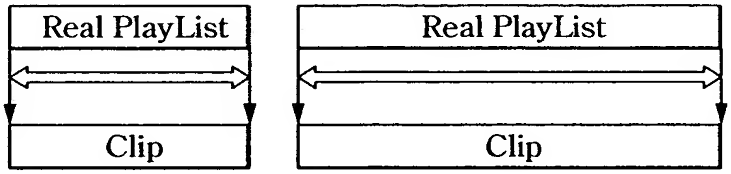


FIG.4A

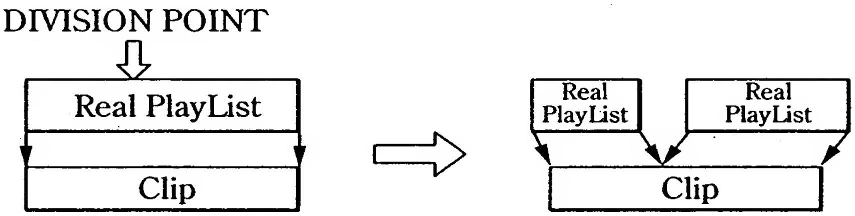


FIG.4B

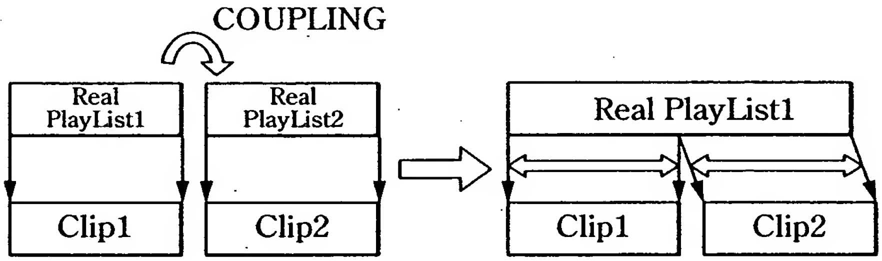


FIG.4C

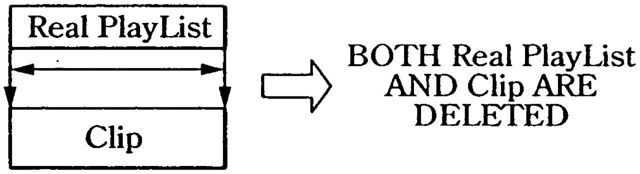


FIG.5A

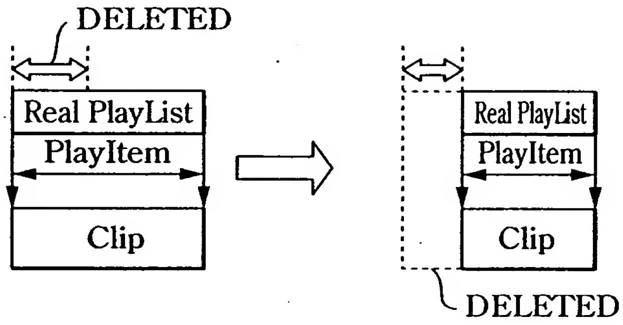


FIG.5B

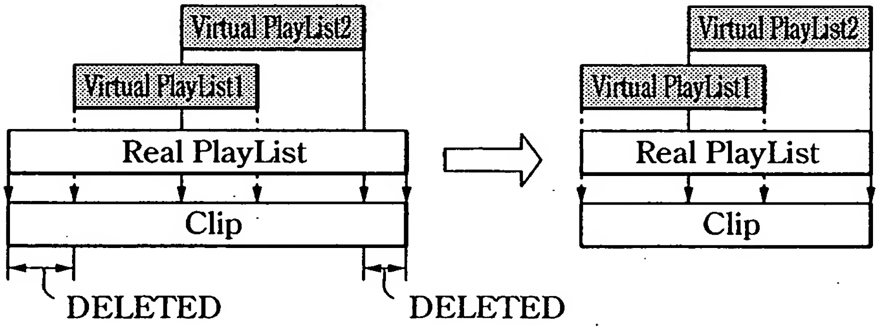


FIG.5C

FIG.6A

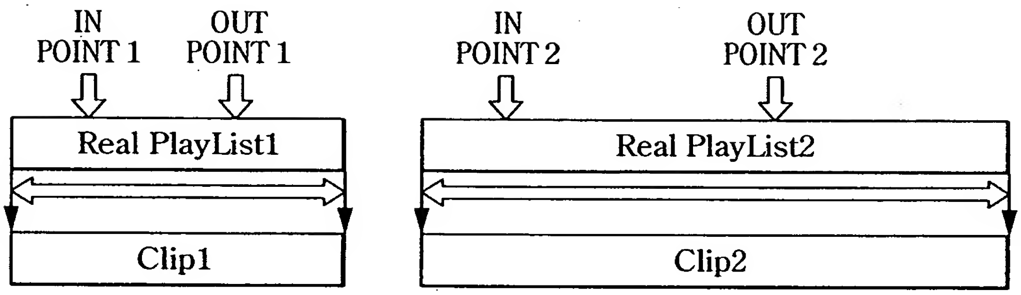
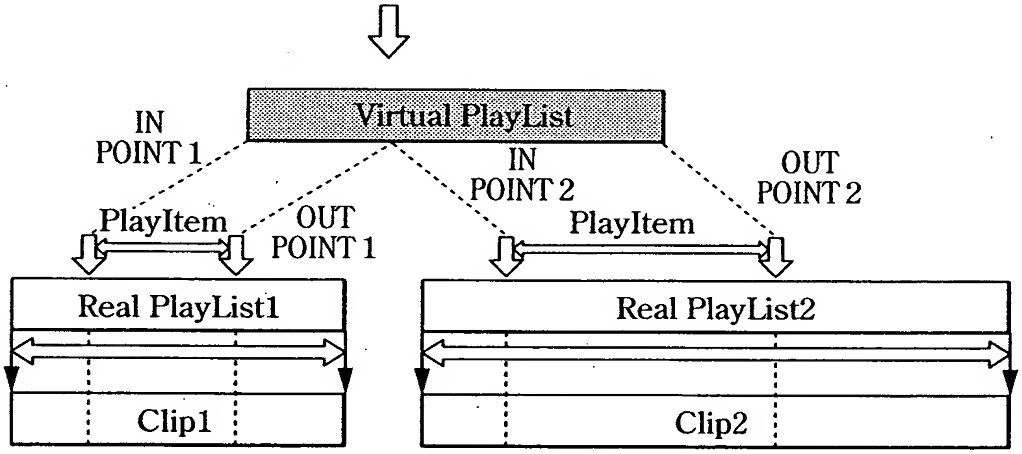

IN
POINT 1
OUT
POINT 1
IN
POINT 2
OUT
POINT 2
Real PlayList1
Real PlayList2
Clip1
Clip2

FIG.6B


Virtual PlayList
IN
POINT 1
PlayItem
OUT
POINT 1
IN
POINT 2
PlayItem
OUT
POINT 2
Real PlayList1
Real PlayList2
Clip1
Clip2

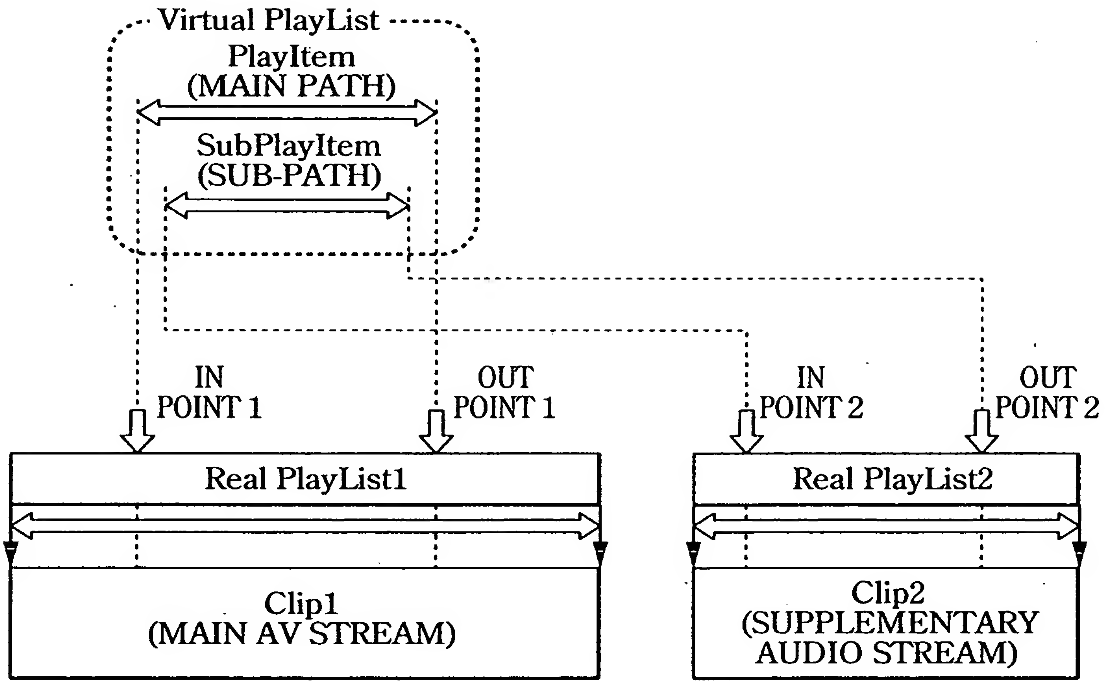

Virtual PlayList
PlayItem
(MAIN PATH)
SubPlayItem
(SUB-PATH)
IN
POINT 1
OUT
POINT 1
IN
POINT 2
OUT
POINT 2
Real PlayList1
Real PlayList2
Clip1
(MAIN AV STREAM)
Clip2
(SUPPLEMENTARY
AUDIO STREAM)

FIG.7

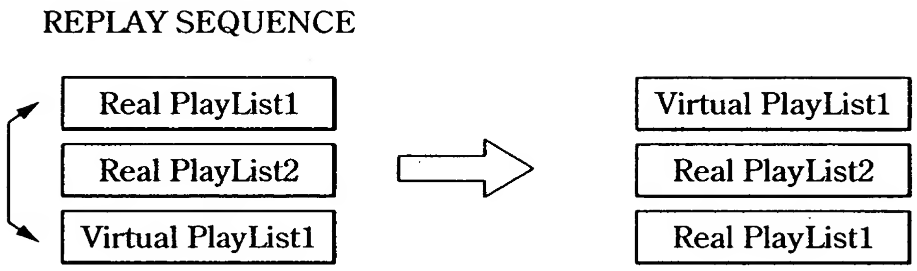

REPLAY SEQUENCE
Real PlayList1
Real PlayList2
Virtual PlayList1
Virtual PlayList1
Real PlayList2
Real PlayList1

FIG.8

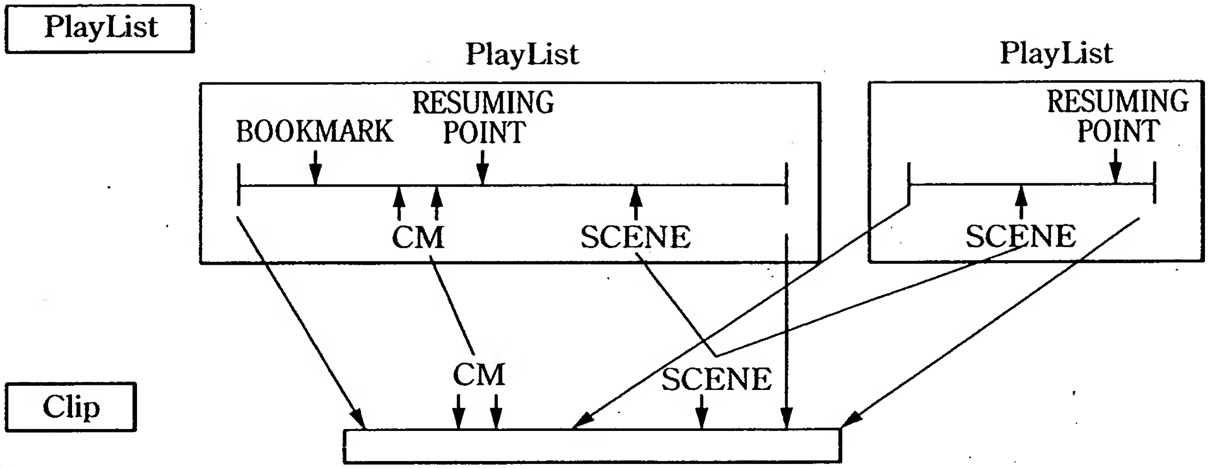

PlayList
PlayList
BOOKMARK
RESUMING POINT
CM
SCENE
PlayList
RESUMING POINT
SCENE
Clip
CM
SCENE

FIG.9

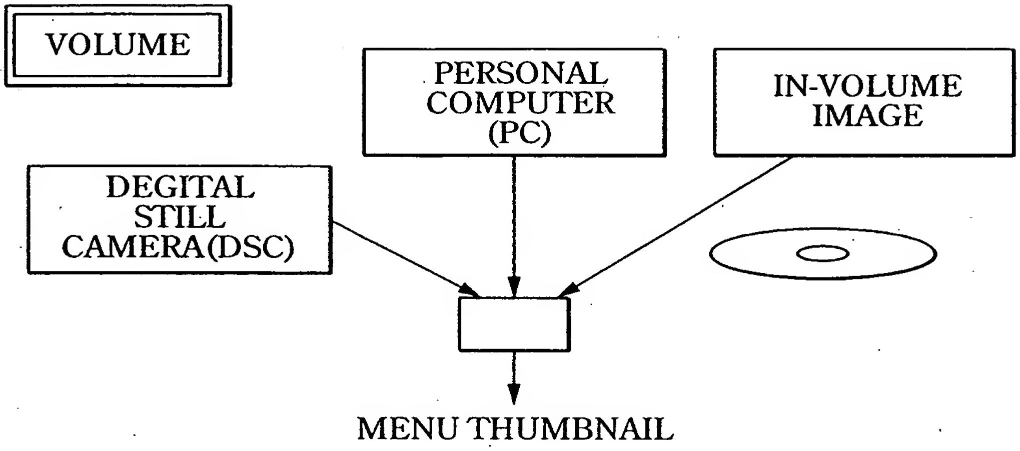

VOLUME
PERSONAL COMPUTER (PC)
IN-VOLUME IMAGE
DEGITAL STILL CAMERA(DSC)
MENU THUMBNAIL

FIG.10

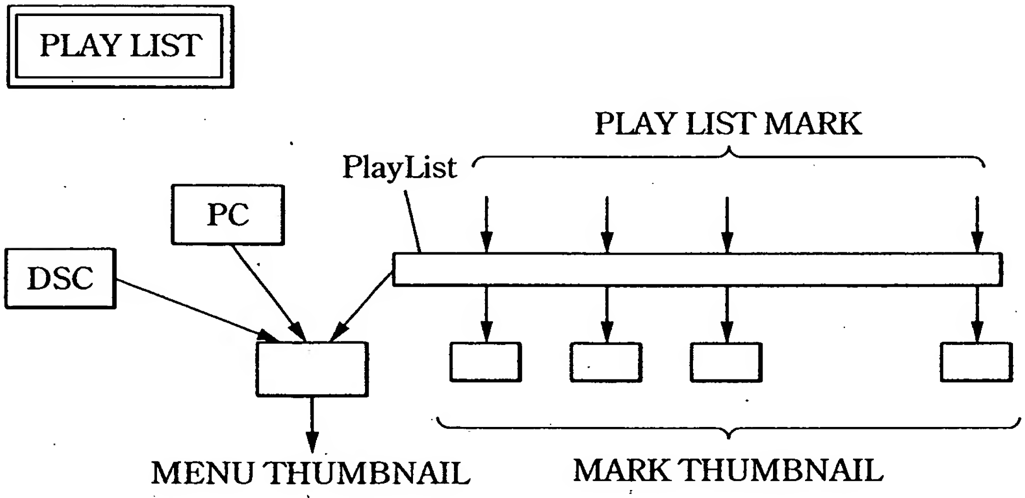

PLAY LIST
PLAY LIST MARK
PlayList
PC
DSC
MENU THUMBNAIL
MARK THUMBNAIL

FIG.11

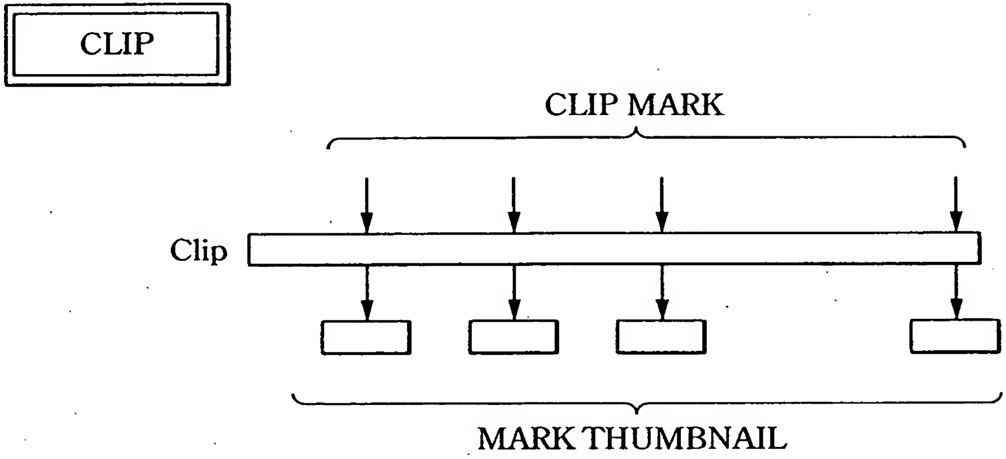

CLIP
CLIP MARK
Clip
MARK THUMBNAIL

FIG.12

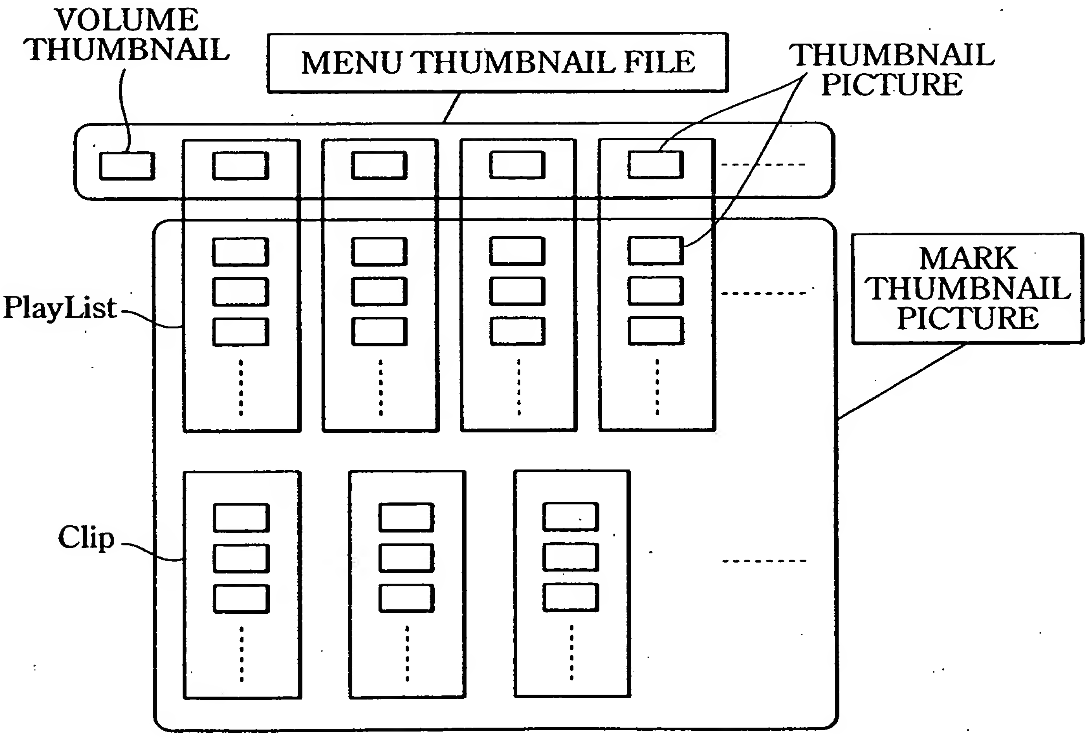

VOLUME THUMBNAIL
MENU THUMBNAIL FILE
THUMBNAIL PICTURE
PlayList
MARK THUMBNAIL PICTURE
Clip

FIG.13

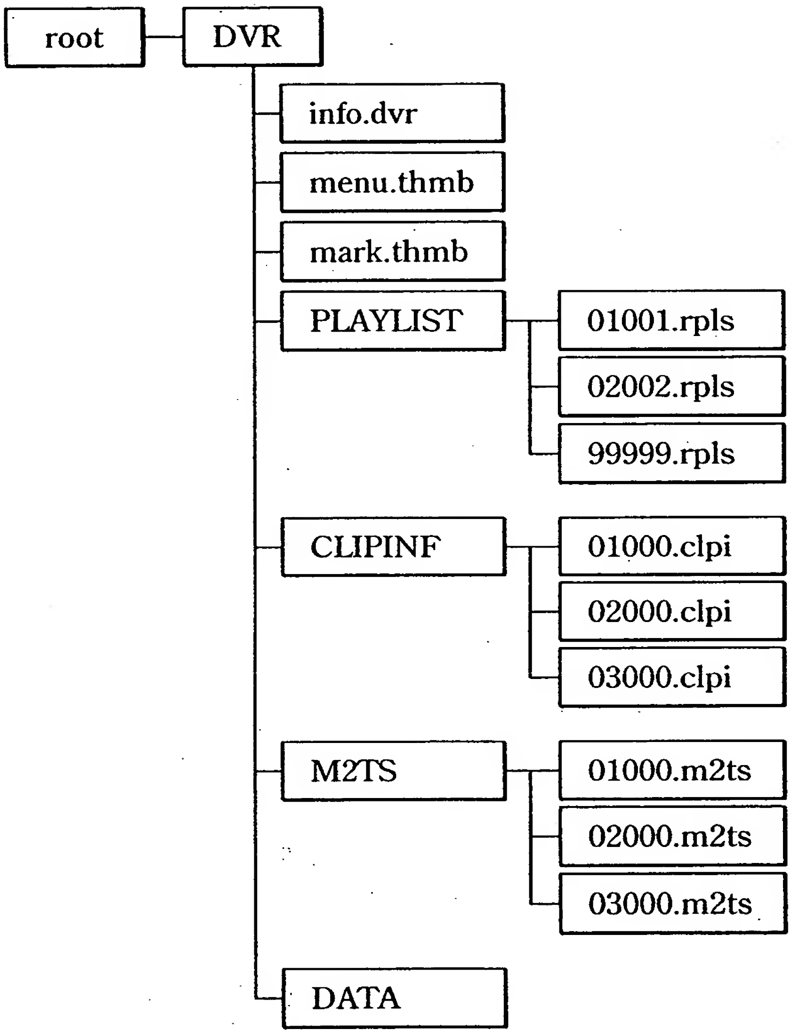

root
DVR
info.dvr
menu.thmb
mark.thmb
PLAYLIST
01001.rpls
02002.rpls
99999.rpls
CLIPINF
01000.clpi
02000.clpi
03000.clpi
M2TS
01000.m2ts
02000.m2ts
03000.m2ts
DATA

FIG.14

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| info.dvr { | | |
| **TableOfPlayLists_Start_address** | 32 | uimsbf |
| **MakersPrivateData_Start_address** | 32 | uimsbf |
| reserved | 192 | bslbf |
| **DVRVolume()** | | |
| for (i=0;i<N1;i++){ | | |
| **padding_word** | 16 | bslbf |
| } | | |
| **TableOfPlayLists()** | | |
| for (i=0;i<N2;i++){ | | |
| **padding_word** | 16 | bslbf |
| } | | |
| **MakersPrivateData()** | | |
| } | | |

FIG.15

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| DVRVolume(){ | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| **ResumeVolume()** | | |
| **UIAppInfoVolume()** | | |
| } | | |

FIG.16

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| ResumeVolume(){ | | |
| reserved | 15 | bslbf |
| **valid_flag** | 1 | bslbf |
| **resume_PlayList_name** | 8*10 | bslbf |
| } | | |

FIG.17

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| UIAppInfoVolume(){ | | |
| **character_set** | 8 | bslbf |
| **name_length** | 8 | uimsbf |
| **Volume_name** | 8*256 | bslbf |
| reserved | 15 | bslbf |
| **Volume_protect_flag** | 1 | bslbf |
| **PIN** | 8*4 | bslbf |
| **ref_thumbnail_index** | 16 | uimsbf |
| **reserved_for_future_use** | 256 | bslbf |
| } | | |

FIG.18

| VALUE | CHARACTER LETTER ENCODING |
|---|---|
| 0x00 | Reserved |
| 0x01 | ISO/IEC 646 (ASCII) |
| 0x02 | ISO/IEC 10646-1 (Unicode) |
| 0x03-0xff | Reserved |

FIG.19

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| TableOfPlayLists () { | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| **number_of_PlayLists** | 16 | uimsbf |
| for (i=0; i<*number_of_PlayLists*; i++) { | | |
| **PlayList_file_name** | 8*10 | bslbf |
| } | | |
| } | | |

FIG.20

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| TableOfPlayLists() { | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| **number_of_PlayLists** | 16 | uimsbf |
| for (i=0; i<*number_of_PlayLists*; i++) { | | |
| **PlayList_file_name** | 8*10 | bslbf |
| **UIAppInfoPlayList()** | | |
| } | | |
| } | | |

FIG.21

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| MakersPrivateData(){ | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| if (length !=0){ | | |
| **mpd_blocks_start_address** | 32 | uimsbf |
| **number_of_maker_entries** | 16 | uimsbf |
| **mpd_block_size** | 16 | uimsbf |
| **number_of_mpd_blocks** | 16 | uimsbf |
| reserved | 16 | bslbf |
| for (i=0; i<*number_of_maker_entries*; i++){ | | |
| **maker_ID** | 16 | uimsbf |
| **maker_model_code** | 16 | uimsbf |
| **start_mpd_block_number** | 16 | uimsbf |
| reserved | 16 | bslbf |
| **mpd_length** | 32 | uimsbf |
| } | | |
| **stuffing_bytes** | 8*2*L1 | bslbf |
| for(j=0; j<*number_of_mpd_blocks*; j++){ | | |
| **mpd_block** | mpd_block_ size*1024*8 | |
| } | | |
| } | | |
| } | | |

FIG.22

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| xxxxx.rpls / yyyyy.vpls { | | |
| **PlayListMark_Start_address** | 32 | uimsbf |
| **MakersPrivateData_Start_address** | 32 | uimsbf |
| reserved | 192 | bslbf |
| **PlayList()** | | |
| for (i=0;i<N1;i++){ | | |
| **padding_word** | 16 | bslbf |
| } | | |
| **PlayListMark()** | | |
| for (i=0;i<N2;i++){ | | |
| **padding_word** | 16 | bslbf |
| } | | |
| **MakersPrivateData()** | | |
| } | | |

FIG.23

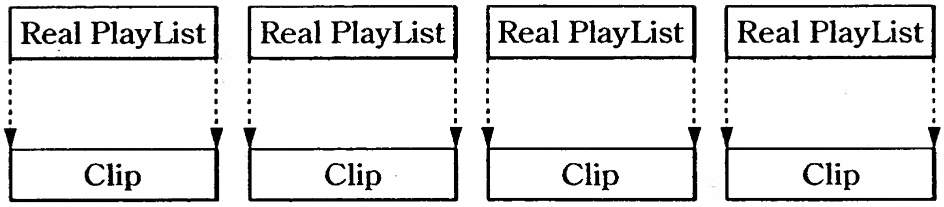


FIG.24A

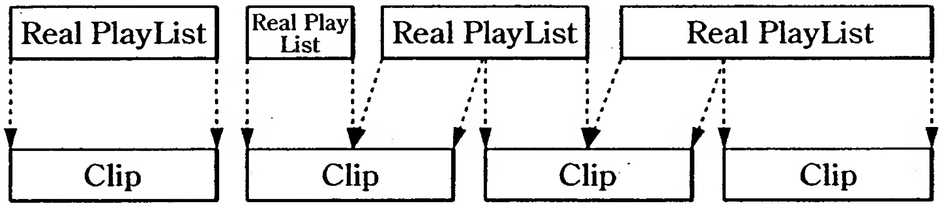


FIG.24B

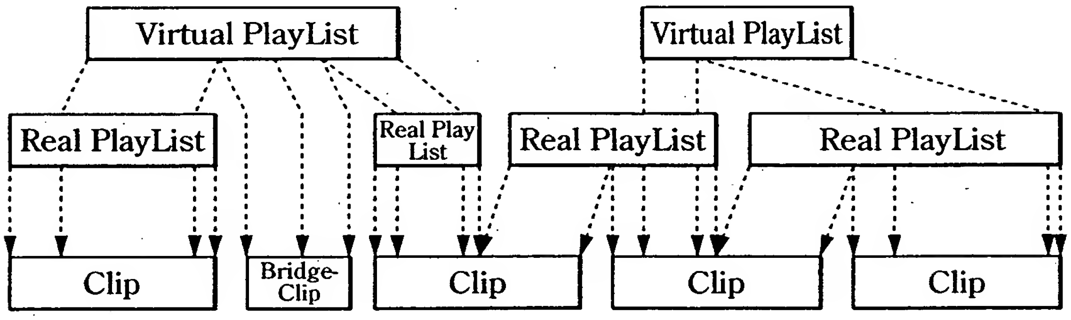


FIG.24C

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| PlayList(){ | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| **PlayList_type** | 8 | uimsbf |
| **CPI_type** | 1 | bslbf |
| reserved | 7 | bslbf |
| **UIAppInfoPlayList()** | | |
| **number_of_PlayItems** // main path | 16 | uimsbf |
| if (<*Virtual PlayList*>){ | | |
| **number_of_SubPlayItems** // sub path | 16 | uimsbf |
| }else{ | | |
| reserved | 16 | bslbf |
| } | | |
| for (*PlayItem_id=0;*<br>*PlayItem_id<nymber_of_PlayItems;*<br>*PlayItem_id++*){ | | |
| **PlayItem()** //main path | | |
| } | | |
| if (<*Virtual PlayList*>){ | | |
| if (CPI_type==0 && PlayList_type==0){ | | |
| for (i=0; i<*number_of _SubPlayItems*; i++) | | |
| **SubPlayItem()** //sub path | | |
| } | | |
| } | | |
| } | | |

FIG.25

| PlayList_type | MEANING |
|---|---|
| 0 | PLAY LIST FOR AV RECORDING<br>ALL CLIPS REFERENCED IN THIS PLAY LIST MUST CONTAIN ONE OR MORE VIDEO STREAMS |
| 1 | PLAY LIST FOR AUDIO RECORDING<br>ALL CLIPS REFERENCED IN THIS PLAYLIST MUST CONTAIN ONE OR MORE AUDIO STREAMS AND MUST NOT CONTAIN VIDEO STREAMS |
| 2-255 | reserved |

FIG.26

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| UIAppInfoPlayList2(){ | | |
| **character_set** | 8 | bslbf |
| **name_length** | 8 | uimsbf |
| **PlayList_name** | 8*256 | bslbf |
| reserved | 8 | bslbf |
| **record_time_and_date** | 4*14 | bslbf |
| reserved | 8 | bslbf |
| **duration** | 4*6 | bslbf |
| **valid_period** | 4*8 | bslbf |
| **maker_id** | 16 | uimsbf |
| **maker_code** | 16 | uimsbf |
| reserved | 11 | bslbf |
| **playback_control_flag** | 1 | bslbf |
| **write_protect_flag** | 1 | bslbf |
| **is_played_flag** | 1 | bslbf |
| **archive** | 2 | bslbf |
| **ref_thumbnail_index** | 16 | uimsbf |
| **reserved_for_future_use** | 256 | bslbf |
| } | | |

**FIG.27**

| write_protect_flag | MEANING |
|---|---|
| 0b | THE PlayList CAN BE ERASED FREELY |
| 1b | THE PlayList CONTENTS SHOULD NOT BE ERASED NOR CHANGED EXCEPT write-protect-flag |

FIG.28A

| is_played_flag | MEANING |
|---|---|
| 0b | THE PlayList HAS NOT BEEN REPRODUCED SINCE ITS RECORDING |
| 1b | THE PlayList WAS ONCE REPRODUCED SINCE ITS RECORDING |

FIG.28B

| archive | MEANING |
|---|---|
| 00b | NO MEANING DEFINED |
| 01b | ORIGINAL |
| 10b | COPY |
| 11b | reserved |

FIG.28C

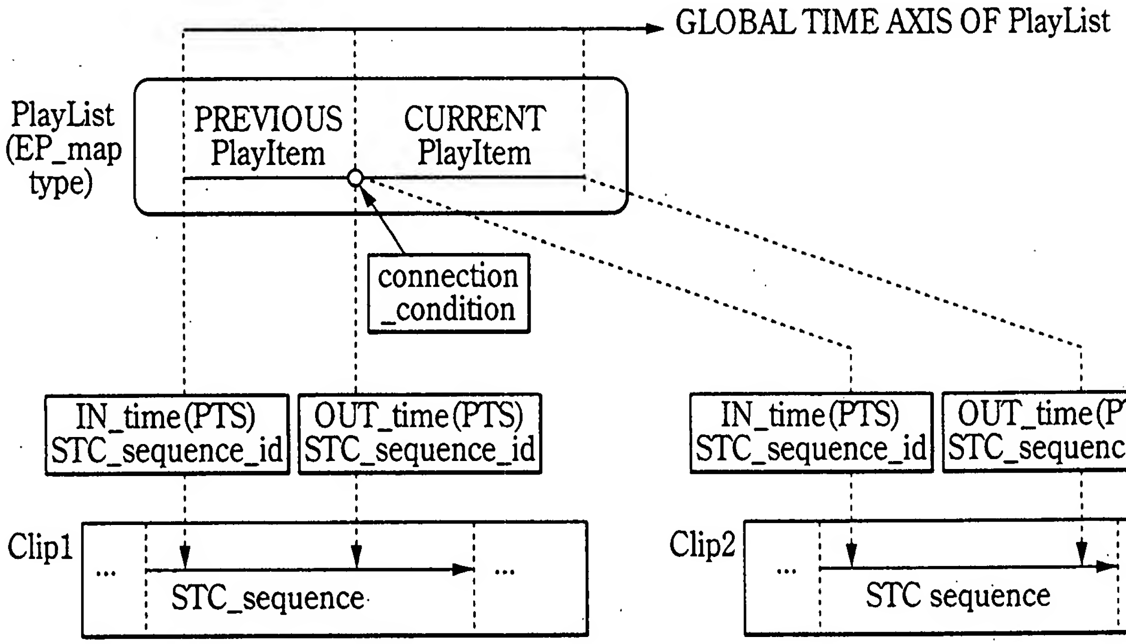

GLOBAL TIME AXIS OF PlayList
PlayList (EP_map type)
PREVIOUS PlayItem
CURRENT PlayItem
connection _condition
IN_time(PTS) STC_sequence_id
OUT_time(PTS) STC_sequence_id
IN_time(PTS) STC_sequence_id
OUT_time(PTS) STC_sequence_id
Clip1
...
STC_sequence
...
Clip2
...
STC sequence
...

FIG.29

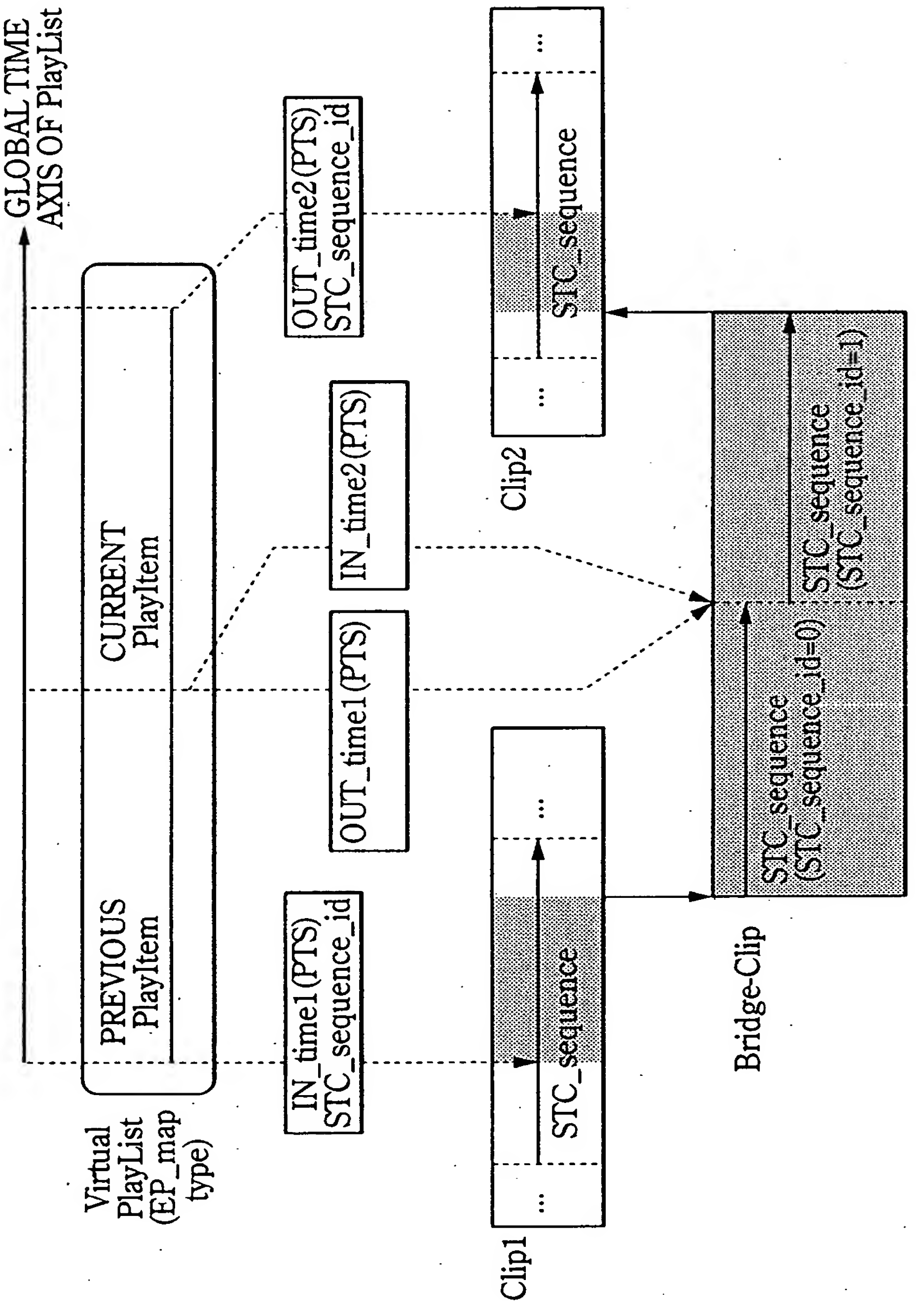

GLOBAL TIME AXIS OF PlayList
Virtual PlayList (EP_map type)
PREVIOUS PlayItem
CURRENT PlayItem
IN_time1(PTS) STC_sequence_id
OUT_time1(PTS)
IN_time2(PTS)
OUT_time2(PTS) STC_sequence_id
Clip1
...
STC_sequence
...
Clip2
...
STC_sequence
...
Bridge-Clip
STC_sequence (STC_sequence_id=0)
STC_sequence (STC_sequence_id=1)

FIG.30

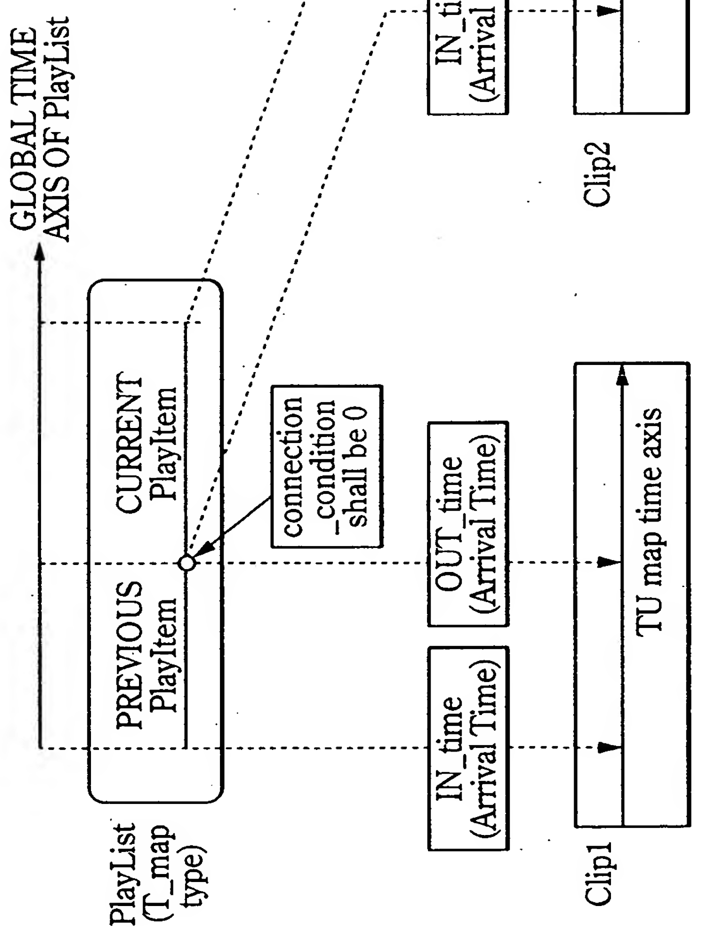

GLOBAL TIME AXIS OF PlayList
PlayList (T_map type)
PREVIOUS PlayItem
CURRENT PlayItem
connection _condition shall be 0
IN_time (Arrival Time)
OUT_time (Arrival Time)
IN_time (Arrival Time)
OUT_time (Arrival Time)
Clip1
TU map time axis
Clip2
TU map time axis

FIG.31

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| PlayItem() { | | |
| **Clip_information_file_name** | 8*10 | bslbf |
| reserved | 24 | bslbf |
| **STC_sequence_id** | 8 | uimsbf |
| **IN_time** | 32 | uimsbf |
| **OUT_time** | 32 | uimsbf |
| reserved | 14 | bslbf |
| **connection_condition** | 2 | bslbf |
| if (<Virtual PlayList>) { | | |
| if (*connection_condition=='10'*) { | | |
| **BridgeSequenceInfo()** | | |
| } | | |
| } | | |
| } | | |

FIG.32

| CPI_type<br>in the PlayList() | SEMANTICS OF IN_time |
|---|---|
| EP_map type | IN_time MUST INDICATE UPPER 32 BITS OF 33 BIT LENGTH CORRESPONDING TO FIRST PRESENTATION UNIT IN PlayItem |
| TU_map type | IN_time MUST BE TIME ON TU_map_time_axis, AND MUST BE ROUNDED TO time_unit PRECISION. IN-time IS CALCULATED BY FOLLOWING EQUATION:<br>IN_time = TU_start_time %$2^{32}$ |

FIG.33

| CPI_type<br>in the PlayList() | SEMANTICS OF OUT_time |
|---|---|
| EP_map type | OUT_time MUST INDICATE UPPER 32 BITS OF THE VALUE OF Presentation_end_TS CALCULATED BY FOLLOWING EQUTION:<br>$\text{Presentation\_end\_TS} = \text{PTS\_out}+\text{AU\_duration}$<br>WHERE PTS_out IS 33-BIT LONG PTS CORRESPONDING TO LAST PRESENTATION UNIT IN PlayItem. AU_duration IS 90 kHz-DISPLAY TIME OF LAST PRESENTATION UNIT. |
| TU_map type | OUT_time MUST BE TIME ON TU_map_time_axis AND BE ROUNDED TO time_unit PRECISION. OUT_time IS CALCULATED BY FOLLOWING EQUATION:<br>$\text{OUT\_time} = \text{TU\_start\_time}\ \%2^{32}$ |

FIG.34

| connection _condition | MEANING |
|---|---|
| 00 | · CONNECTION OF PREVIOUS PlayItem TO CURRENT PlayItem IS NOT SURE AS TO SEAMLESS REPLAY.<br>· IF CPI_type OF PlayList IS TU_map type, THIS VALUE MUST BE SET IN connection_condition. |
| 01 | · THIS STATE IS ALLOWED ONLY WHEN CPI_type OF PlayList IS EP_map type.<br>· PREVIOUS PlayItem AND CURRENT PlayItem INDICATE DIVISION BECAUSE OF NON-CONTINUOUS POINT OF SYSTEM TIMEBASE (STC BASE). |
| 10 | · THIS STATE IS ALLOWED ONLY WHEN CPI_type OF PlayList IS EP_map type.<br>· THIS STATE IS ALLOWED ONLY FOR Virtual PlayList.<br>· CONNECTION OF PREVIOUS PlayItem TO CURRENT PlayItem IS SURE AS TO SEAMLESS REPLAY.<br>· PREVIOUS PlayItem IS CONNECTED TO CURRENT PlayItem USING BridgeSequence. DVR MPEG-2 TRANSPORT STREAM MUST OBEY DVR-STD AS LATER DESCRIBED. |
| 11 | · THIS STATE IS ALLOWED ONLY WHEN CPI_type OF PlayList IS EP_map type.<br>· CONNECTION OF PREVIOUS PlayItem TO CURRENT Play Item IS SURE AS TO SEAMLESS REPLAY.<br>· PREVIOUS PlayItem IS CONNECTED TO CURRENT PlayItem WITHOUT USING BridgeSequence. DVR MPEG-2 TRANSPORT STREAM MUST OBEY DVR-STD AS LATER DESCRIBED. |

FIG.35

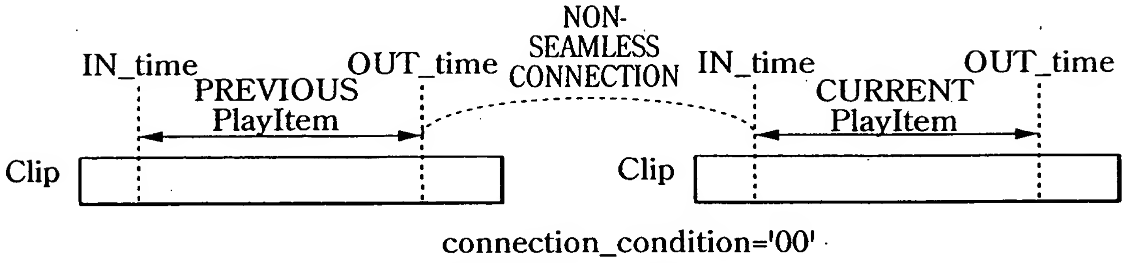


FIG.36A

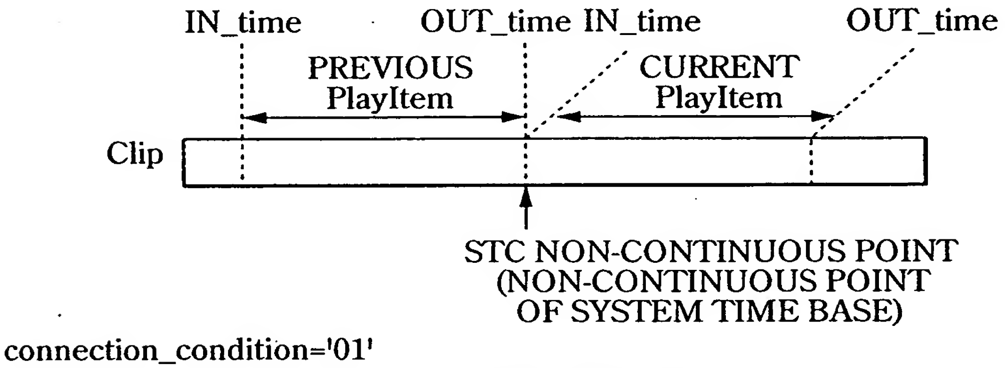

IN_time
OUT_time
IN_time
OUT_time
PREVIOUS PlayItem
CURRENT PlayItem
Clip
STC NON-CONTINUOUS POINT (NON-CONTINUOUS POINT OF SYSTEM TIME BASE)
connection_condition='01'

FIG.36B

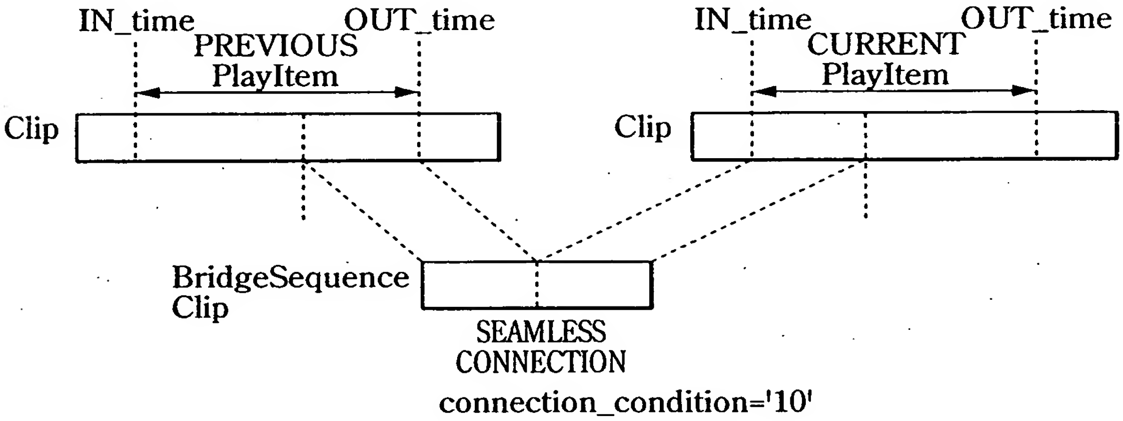


FIG.36C

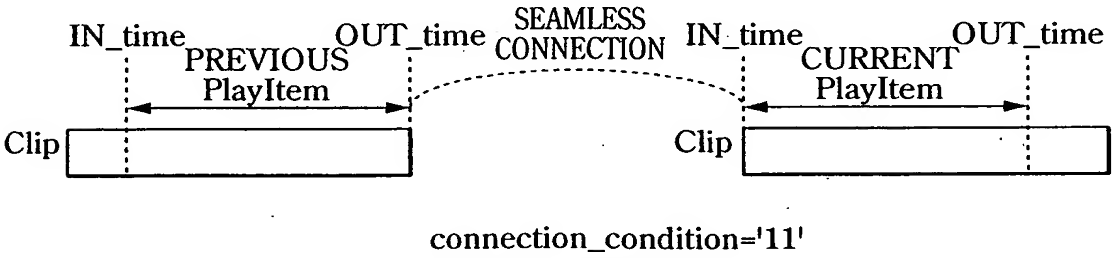


FIG.36D

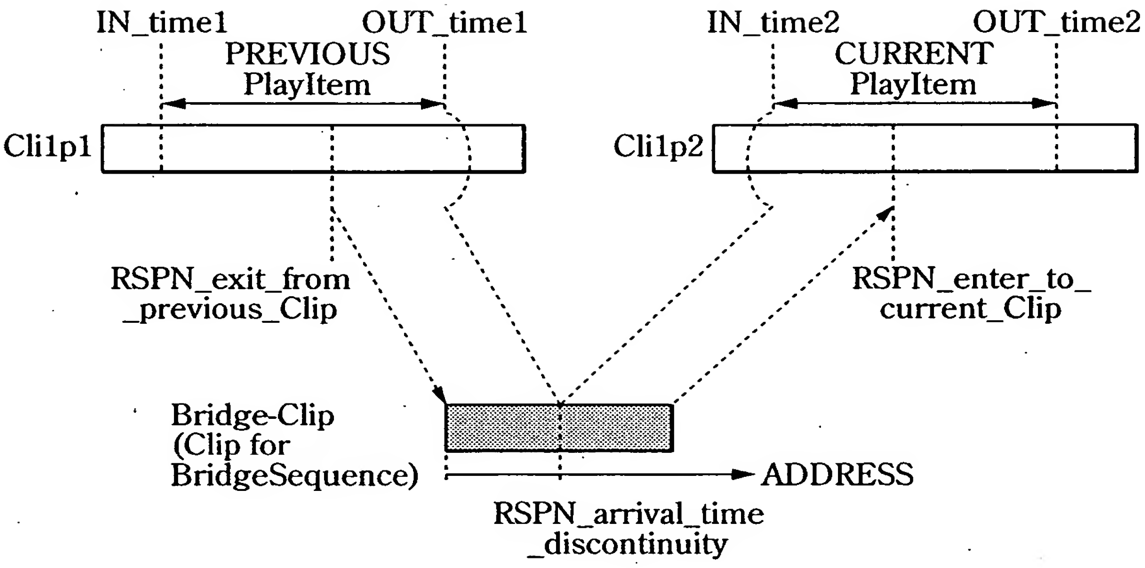

IN_time1
OUT_time1
PREVIOUS
PlayItem
IN_time2
OUT_time2
CURRENT
PlayItem
Cli1p1
Cli1p2
RSPN_exit_from
_previous_Clip
RSPN_enter_to_
current_Clip
Bridge-Clip
(Clip for
BridgeSequence)
ADDRESS
RSPN_arrival_time
_discontinuity

FIG.37

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| BridgeSequenceInfo() { | | |
| **Bridge_Clip_information_file_name** | 8*10 | bslbf |
| **RSPN_exit_from_previous_Clip** | 32 | uimsbf |
| **RSPN_enter_to_current_Clip** | 32 | uimsbf |
| } | | |

FIG.38

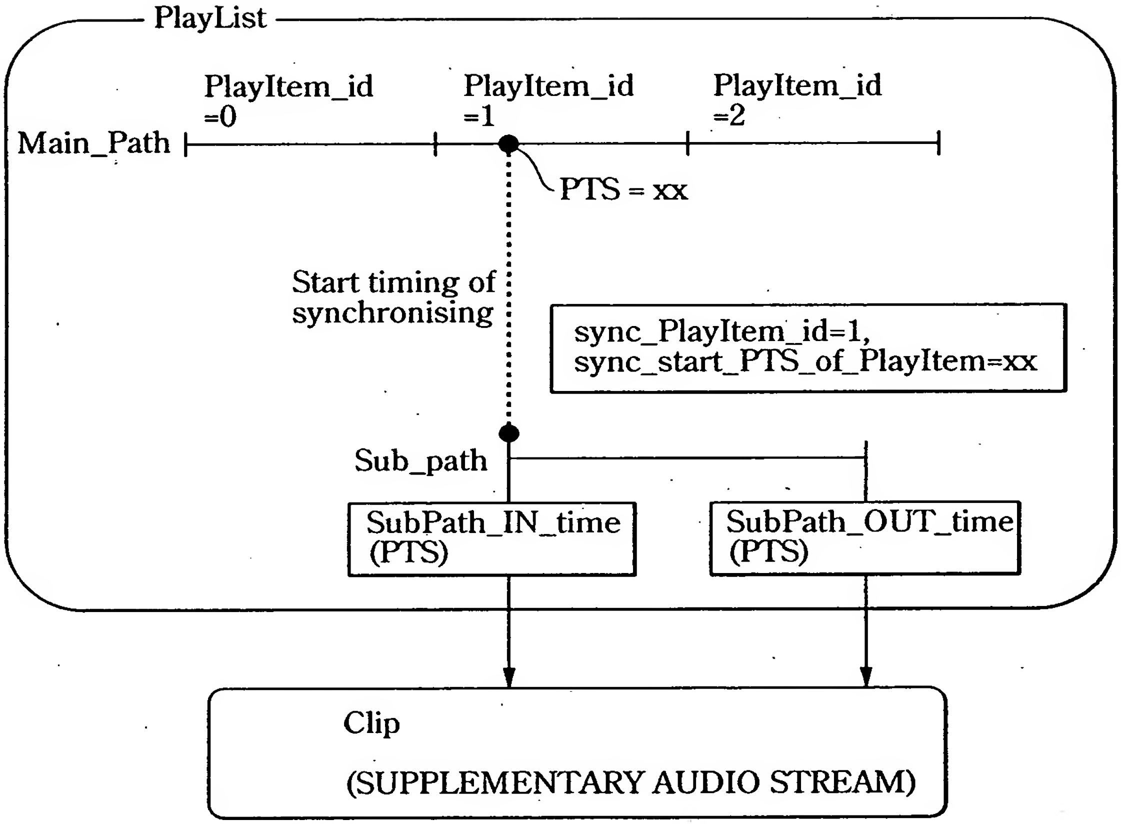

PlayList
PlayItem_id =0
PlayItem_id =1
PlayItem_id =2
Main_Path
PTS = xx
Start timing of synchronising
sync_PlayItem_id=1,
sync_start_PTS_of_PlayItem=xx
Sub_path
SubPath_IN_time (PTS)
SubPath_OUT_time (PTS)
Clip
(SUPPLEMENTARY AUDIO STREAM)

FIG.39

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| SubPlayItem(){ | | |
| **Clip_Information_file_name** | 8*10 | bslbf |
| **SubPath_type** | 8 | bslbf |
| **sync_PlayItem_id** | 8 | uimsbf |
| **sync_start_PTS_of_PlayItem** | 32 | uimsbf |
| **SubPath_IN_time** | 32 | uimsbf |
| **SubPath_OUT_time** | 32 | uimsbf |
| } | | |

FIG.40

| SubPath_type | MEANING |
|---|---|
| 0x00 | Auxiliary audio steam path |
| 0x01-0xff | reserved |

FIG.41

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| PlayListMark() { | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| **number_of_PlayList_marks** | 16 | uimsbf |
| for (i=0;i<*number_of_PlayList_marks*;i++) { | | |
| reserved | 8 | bslbf |
| **mark_type** | 8 | bslbf |
| **mark_time_stamp** | 32 | uimsbf |
| **PlayItem_id** | 8 | uimsbf |
| reserved | 24 | uimsbf |
| **character_set** | 8 | bslbf |
| **name_length** | 8 | uimsbf |
| **mark_name** | 8*256 | bslbf |
| **ref_thumbnail_index** | 16 | uimsbf |
| } | | |
| } | | |

FIG.42

| Mark_type | MEANING | COMMENT |
|---|---|---|
| 0x00 | resume-mark | REPLAY RESUME POINT. THE NUMBER OF REPLAY RESURE POINTS DEFINED IN PlayListMark() MUST BE 0 OR 1. |
| 0x01 | book-mark | REPLAY ENTRY POINT OF PlayList. THIS MARK CAN BE SET BY USER AND USED AS MARK SPECIFYING START POINT OF FAVORITE SCENE. |
| 0x02 | skip-mark | SKIP MARK POINT. PLAYER SKIPS PROGRAM FROM THIS POINT TO THE END OF PROGRAM. THE NUMBER OF SKIP MARK POINTS DEFINED IN PlayListMark() MUST BE 0 RO 1. |
| 0x03-0x8F | reserved | |
| 0x90-0xFF | reserved | Reserved for ClipMark() |

FIG.43

| CPI_type<br>in the PlayList() | SEMANTICS OF mark_time_stamp |
|---|---|
| EP_map type | mark_time_stamp MUST INDICATE UPPER 32 BITS OF 33 BIT LENGTH PTS CORRESPONDING TO PRESENTATION UNIT REFERENCED BY MARK. |
| TU_map type | mark_time_stamp MUST BE TIME ON TU_map_time_axis AND MUST BE ROUNDED TO time_unit PRECISION. mark_time_stamp IS CALCULATED BY FOLLOWING EQUATION:<br>mark_time_stamp = TU_start_time %$2^{32}$ |

FIG.44

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| zzzzz.clpi { | | |
| **STC_Info_Start_address** | 32 | uimsbf |
| **ProgramInfo_Start_address** | 32 | uimsbf |
| **CPI_Start_address** | 32 | uimsbf |
| **ClipMark_Start_address** | 32 | uimsbf |
| **MakersPrivateData_Start_address** | 32 | uimsbf |
| reserved | 96 | bslbf |
| **ClipInfo()** | | |
| for (i=0;i<N1;i++){ | | |
| **padding_word** | 16 | bslbf |
| } | | |
| **STC_Info()** | | |
| for (i=0;i<N2;i++){ | | |
| **padding_word** | 16 | bslbf |
| } | | |
| **ProgramInfo()** | | |
| for (i=0;i<N3;i++){ | | |
| **padding_word** | 16 | bslbf |
| } | | |
| **CPI()** | | |
| for (i=0;i<N4;i++){ | | |
| **padding_word** | 16 | bslbf |
| } | | |
| **ClipMark()** | | |
| for (i=0;i<N5;i++){ | | |
| **padding_word** | 16 | bslbf |
| } | | |
| **MakersPrivateData()** | | |
| } | | |

FIG.45

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| ClipInfo() { | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| **Clip_stream_type** | 8 | bslbf |
| **offset_SPN** | 32 | uimsbf |
| **TS_recording_rate** | 24 | uimsbf |
| reserved | 8 | bslbf |
| **record_time_and_date** | 4*14 | bslbf |
| reserved | 8 | bslbf |
| **duration** | 4*6 | bslbf |
| reserved | 7 | bslbf |
| **time_controlled_flag** | 1 | bslbf |
| **TS_average_rate** | 24 | uimsbf |
| *if (Clip_stream_type==1) // Bridge-Clip AV stream* | | |
| **RSPN_arrival_time_discontinuity** | 32 | uimsbf |
| else | | |
| reserved | 32 | bslbf |
| **reserved_for_system_use** | 144 | bslbf |
| reserved | 11 | bslbf |
| **is_format_identifier_valid** | 1 | bslbf |
| **is_original_network_ID_valid** | 1 | bslbf |
| **is_transport_stream_ID_valid** | 1 | bslbf |
| **is_service_ID_valid** | 1 | bslbf |
| **is_country_code_valid** | 1 | bslbf |
| **format_identifier** | 32 | bslbf |
| **original_network_ID** | 16 | uimsbf |
| **transport_stream_ID** | 16 | uimsbf |
| **service_ID** | 16 | uimsbf |
| **country_code** | 24 | bslbf |
| **stream_format_name** | 16*8 | bslbf |
| **reserved_for_fortune_use** | 256 | bslbf |
| } | | |

FIG.46

| Clip_stream_type | MEANING |
|---|---|
| 0 | Clip AV STREAM |
| 1 | Bridge-Clip AV STREAM |
| 2-255 | Reserved |

FIG.47

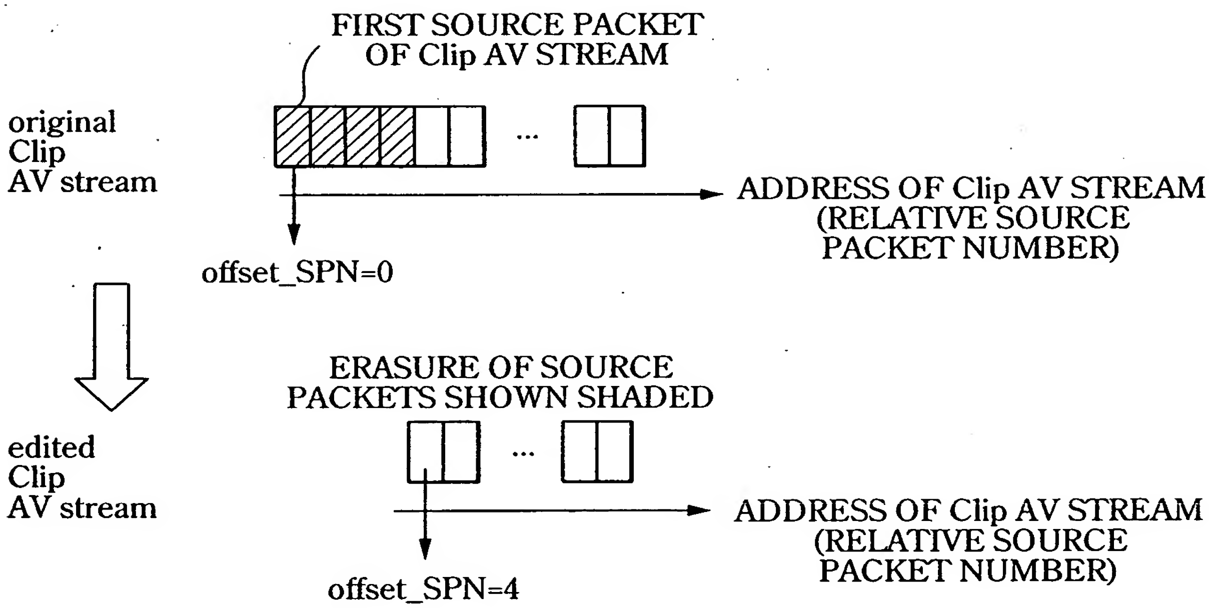

FIRST SOURCE PACKET
OF Clip AV STREAM
original
Clip
AV stream
...
ADDRESS OF Clip AV STREAM
(RELATIVE SOURCE
PACKET NUMBER)
offset_SPN=0
ERASURE OF SOURCE
PACKETS SHOWN SHADED
edited
Clip
AV stream
...
ADDRESS OF Clip AV STREAM
(RELATIVE SOURCE
PACKET NUMBER)
offset_SPN=4

FIG.48

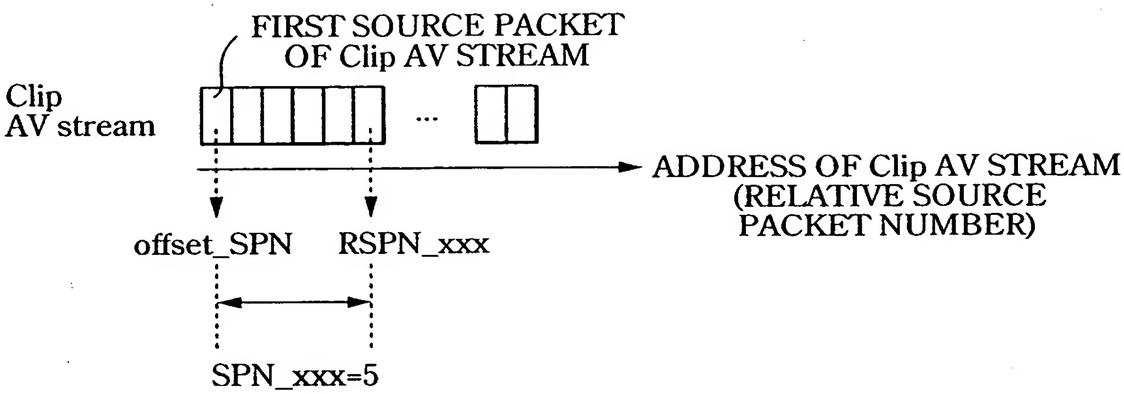


FIG.49

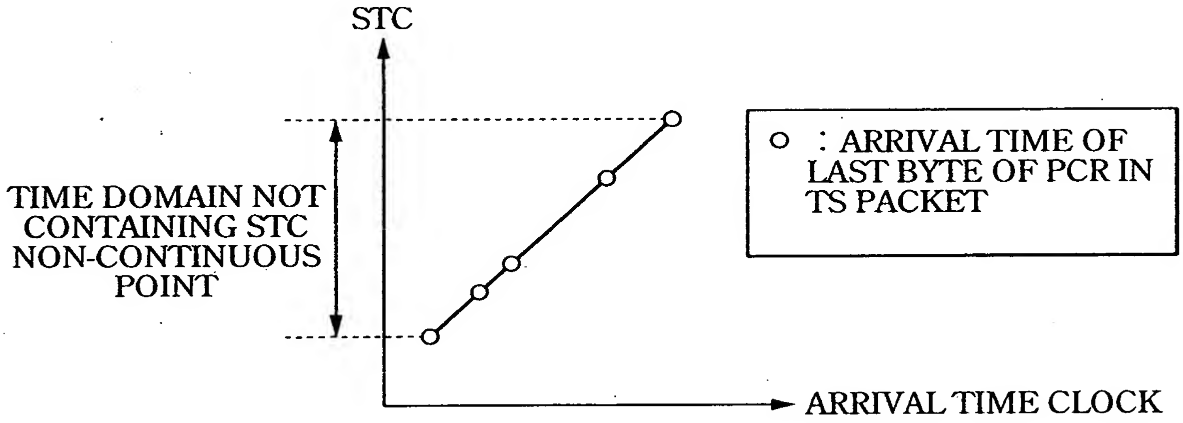


FIG.50A

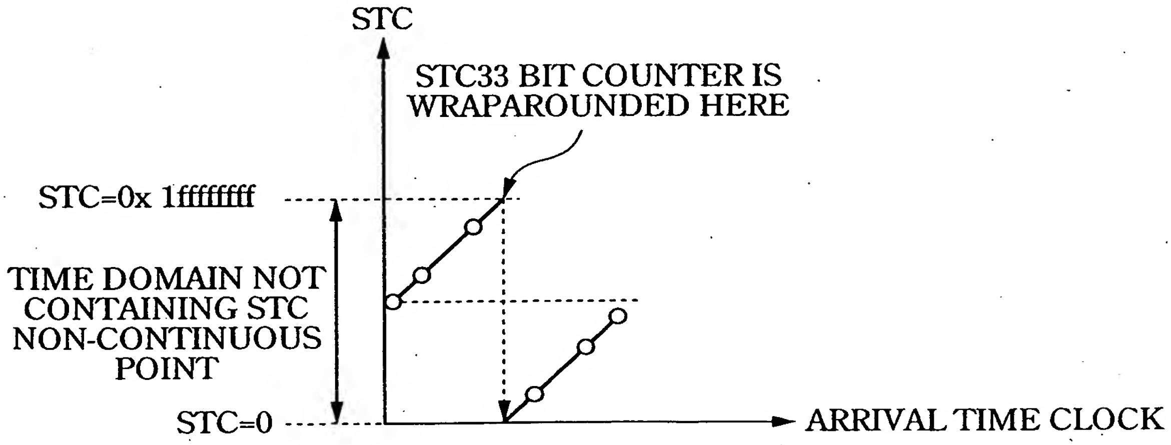


FIG.50B

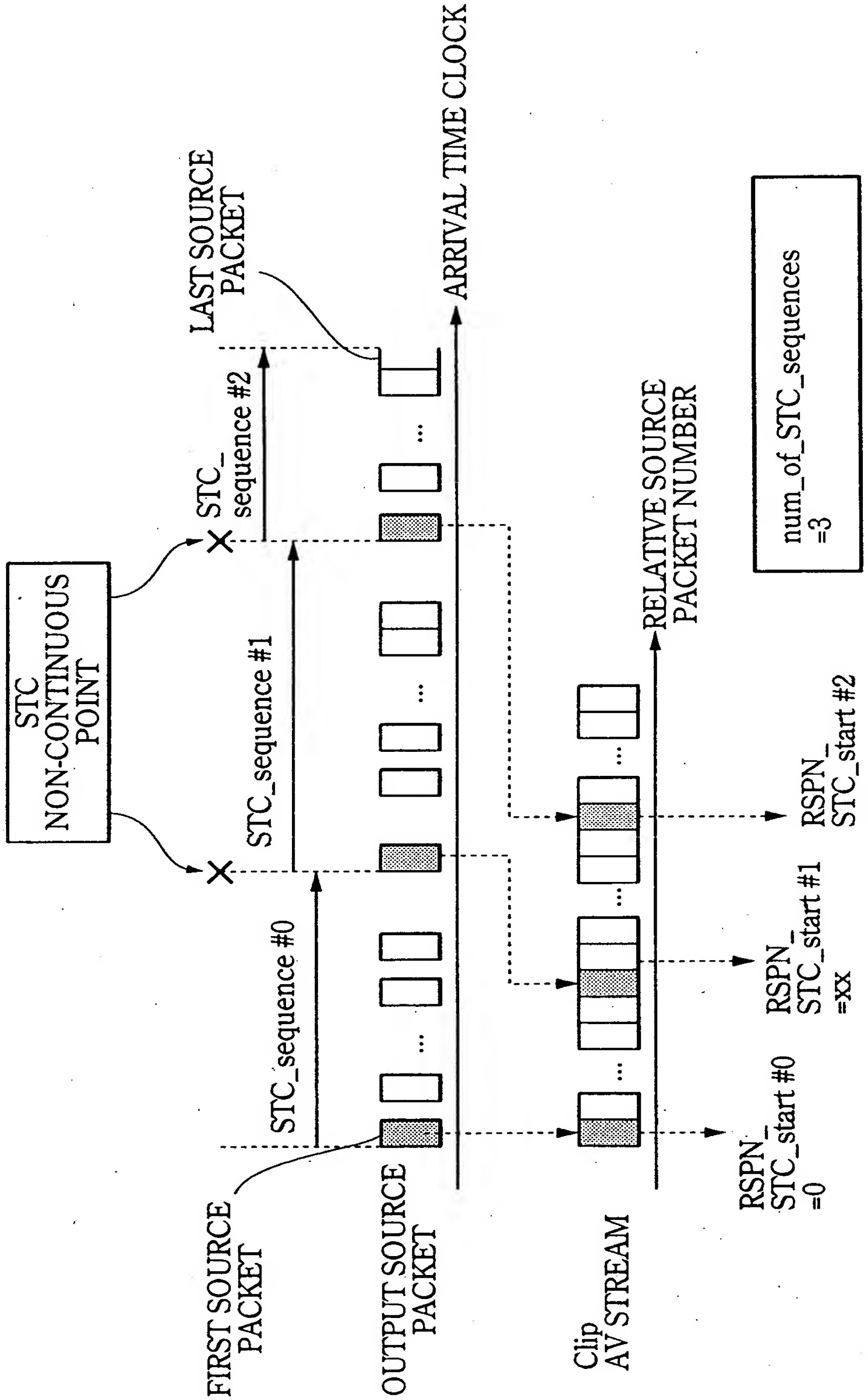

STC NON-CONTINUOUS POINT
LAST SOURCE PACKET
ARRIVAL TIME CLOCK
STC_sequence #0
STC_sequence #1
STC_ sequence #2
FIRST SOURCE PACKET
OUTPUT SOURCE PACKET
Clip AV STREAM
RELATIVE SOURCE PACKET NUMBER
RSPN_ STC_start #0 =0
RSPN_ STC_start #1 =xx
RSPN_ STC_start #2 =yy
num_of_STC_sequences =3

FIG.51

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| STC_Info(){ | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| if (length !=0){ | | |
| reserved | 8 | bslbf |
| **num_of_STC_sequences** | 8 | uimsbf |
| for (*STC_sequence_id=0;*<br>*STC_sequence_id<num_of_STC_sequences;*<br>*STC_sequence_id++*){ | | |
| resereved | 32 | bslbf |
| **RSPN_STC_start** | 32 | uimsbf |
| } | | |
| } | | |
| } | | |

FIG.52

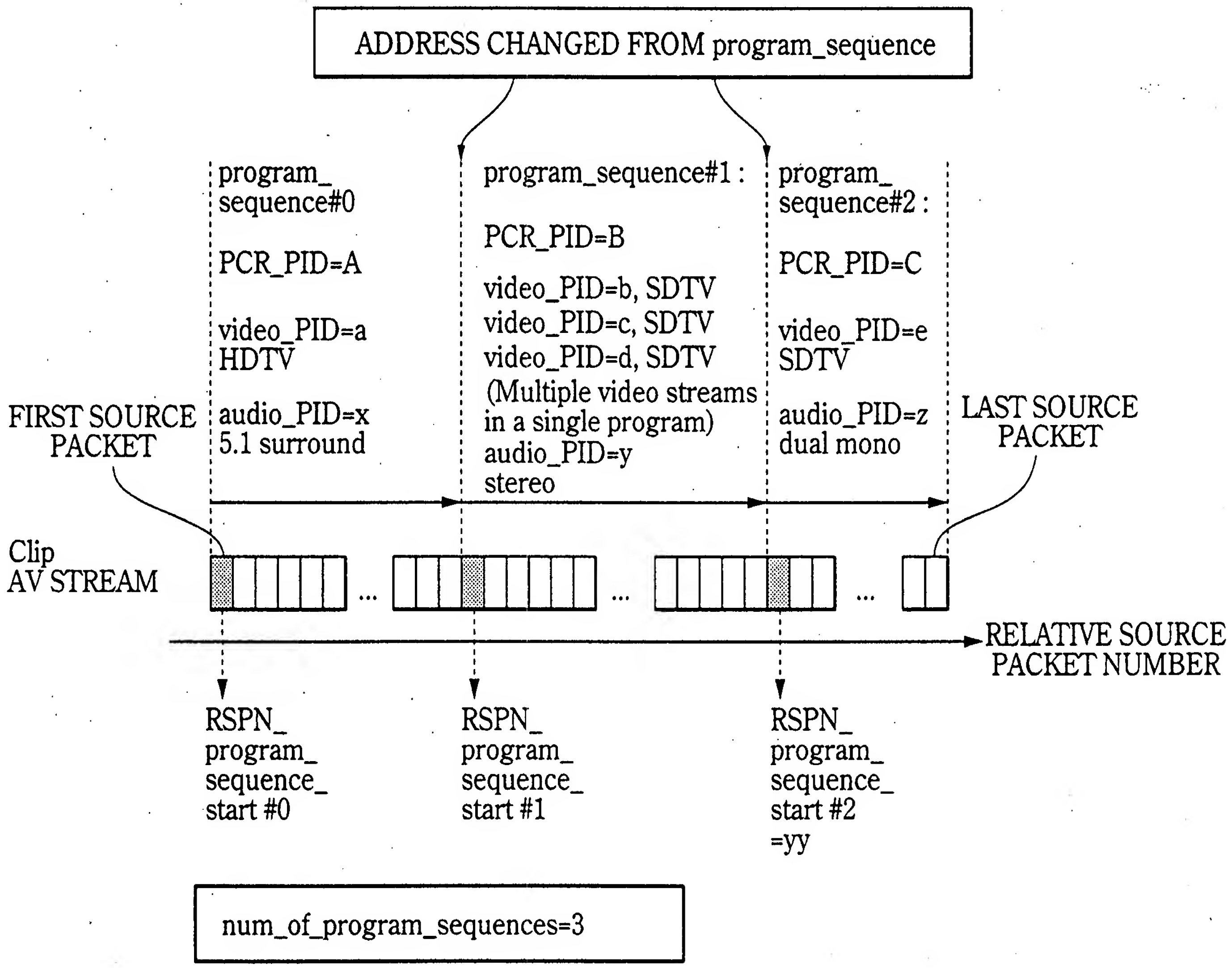

ADDRESS CHANGED FROM program_sequence
program_ sequence#0
PCR_PID=A
video_PID=a HDTV
audio_PID=x 5.1 surround
program_sequence#1 :
PCR_PID=B
video_PID=b, SDTV
video_PID=c, SDTV
video_PID=d, SDTV
(Multiple video streams in a single program)
audio_PID=y stereo
program_ sequence#2 :
PCR_PID=C
video_PID=e SDTV
audio_PID=z dual mono
FIRST SOURCE PACKET
LAST SOURCE PACKET
Clip AV STREAM
...
...
...
RELATIVE SOURCE PACKET NUMBER
RSPN_ program_ sequence_ start #0
RSPN_ program_ sequence_ start #1
RSPN_ program_ sequence_ start #2 =yy
num_of_program_sequences=3

FIG.53

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| ProgramInfo() { | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| if (length !=0) { | | |
| reserved | 8 | bslbf |
| **number_of_program_sequences** | 8 | uimsbf |
| for (i=0;i<*number_of_program_sequences*;i++) { | | |
| **RSPN_program_sequence_start** | 32 | uimsbf |
| reserved | 48 | bslbf |
| **PCR_PID** | 16 | bslbf |
| **number_of_videos** | 8 | uimsbf |
| **number_of_audios** | 8 | uimsbf |
| for (k=0;k<*number_of_videos*;k++) { | | |
| **video_stream_PID** | 16 | bslbf |
| **VideoCodingInfo()** | | |
| } | | |
| for (k=0;k<*number_of_audios*;k++) { | | |
| **audio_stream_PID** | 16 | bslbf |
| **AudioCodingInfo()** | | |
| } | | |
| } | | |
| } | | |
| } | | |

FIG.54

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| VideoCodingInfo() { | | |
| **video_format** | 8 | uimsbf |
| **frame_rate** | 8 | uimsbf |
| **display_aspect_ratio** | 8 | uimsbf |
| reserved | 8 | bslbf |
| } | | |

FIG.55

| video_format | MEANING |
|---|---|
| 0 | 480i |
| 1 | 576i |
| 2 | 480p(including 640×480p format) |
| 3 | 1080i |
| 4 | 720p |
| 5 | 1080p |
| 6-254 | reserved |
| 255 | No information |

FIG.56

| frame_rate | MEANING |
|---|---|
| 0 | forbidden |
| 1 | 24 000/1001 (23.976...) |
| 2 | 24 |
| 3 | 25 |
| 4 | 30 000/1001 (29.97..) |
| 5 | 30 |
| 6 | 50 |
| 7 | 60 000/1001 (59.94..) |
| 8 | 60 |
| 9-254 | reserved |
| 255 | No information |

FIG.57

| display_aspect_ratio | MEANING |
|---|---|
| 0 | forbidden |
| 1 | reserved |
| 2 | 4:3 display aspect ratio |
| 3 | 16:9 display aspect ration |
| 4-254 | reserved |
| 255 | No information |

FIG.58

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| AudioCodingInfo() { | | |
| **audio_format** | 8 | uimsbf |
| **audio_component_type** | 8 | uimsbf |
| **sampling_frequency** | 8 | uimsbf |
| reserved | 8 | bslbf |
| } | | |

FIG.59

| audio_coding | MEANING |
|---|---|
| 0 | MPEG-1 audio layer I or II |
| 1 | Dolby AC-3 audio |
| 2 | MPEG-2 AAC |
| 3 | MPEG-2 multi-channel audio, backward compatible to MPEG-1 |
| 4 | SESF LPCM audio |
| 5-254 | reserved |
| 255 | No information |

FIG.60

| audio_component_type | MEANING |
|---|---|
| 0 | single mono channel |
| 1 | dual mono channel |
| 2 | stereo (2-channel) |
| 3 | multi-lingual, multi-channel |
| 4 | surround sound |
| 5 | audio description for the visually impaired |
| 6 | audio for the hard of hearing |
| 7-254 | reserved |
| 255 | No information |

FIG.61

| sampling_frequency | MEANING |
|---|---|
| 0 | 48 kHz |
| 1 | 44.1 kHz |
| 2 | 32 kHz |
| 3-254 | reserved |
| 255 | No information |

FIG.62

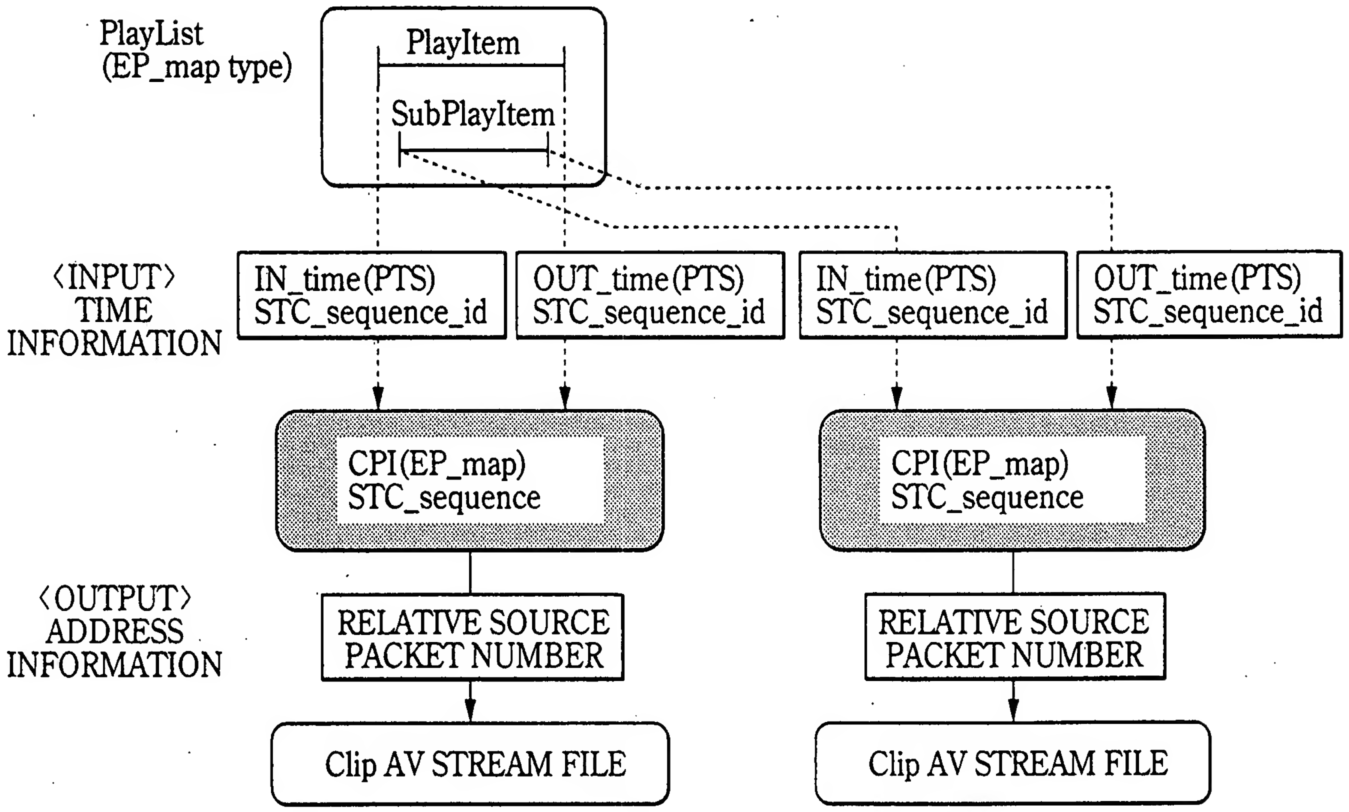

PlayList
(EP_map type)
PlayItem
SubPlayItem
〈INPUT〉
TIME
INFORMATION
IN_time(PTS)
STC_sequence_id
OUT_time(PTS)
STC_sequence_id
IN_time(PTS)
STC_sequence_id
OUT_time(PTS)
STC_sequence_id
CPI(EP_map)
STC_sequence
CPI(EP_map)
STC_sequence
〈OUTPUT〉
ADDRESS
INFORMATION
RELATIVE SOURCE
PACKET NUMBER
RELATIVE SOURCE
PACKET NUMBER
Clip AV STREAM FILE
Clip AV STREAM FILE

FIG.63

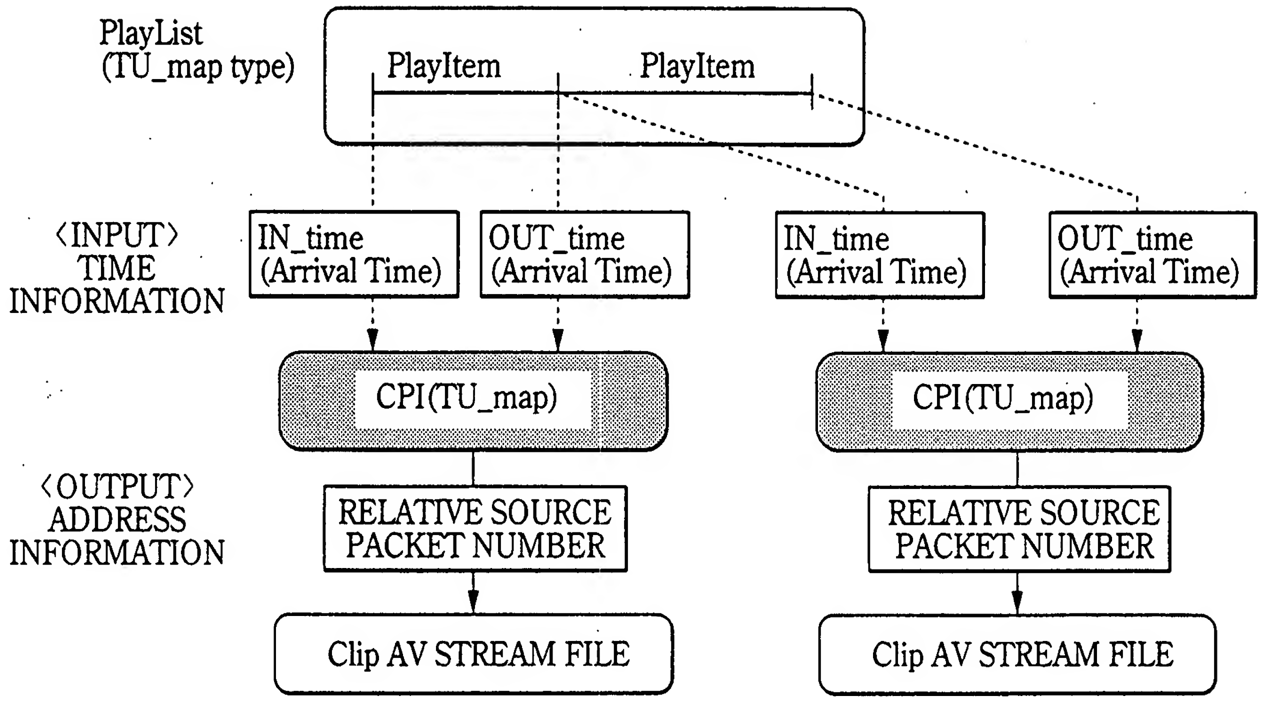

PlayList
(TU_map type)
PlayItem
PlayItem
〈INPUT〉
TIME
INFORMATION
IN_time
(Arrival Time)
OUT_time
(Arrival Time)
IN_time
(Arrival Time)
OUT_time
(Arrival Time)
CPI(TU_map)
CPI(TU_map)
〈OUTPUT〉
ADDRESS
INFORMATION
RELATIVE SOURCE
PACKET NUMBER
RELATIVE SOURCE
PACKET NUMBER
Clip AV STREAM FILE
Clip AV STREAM FILE

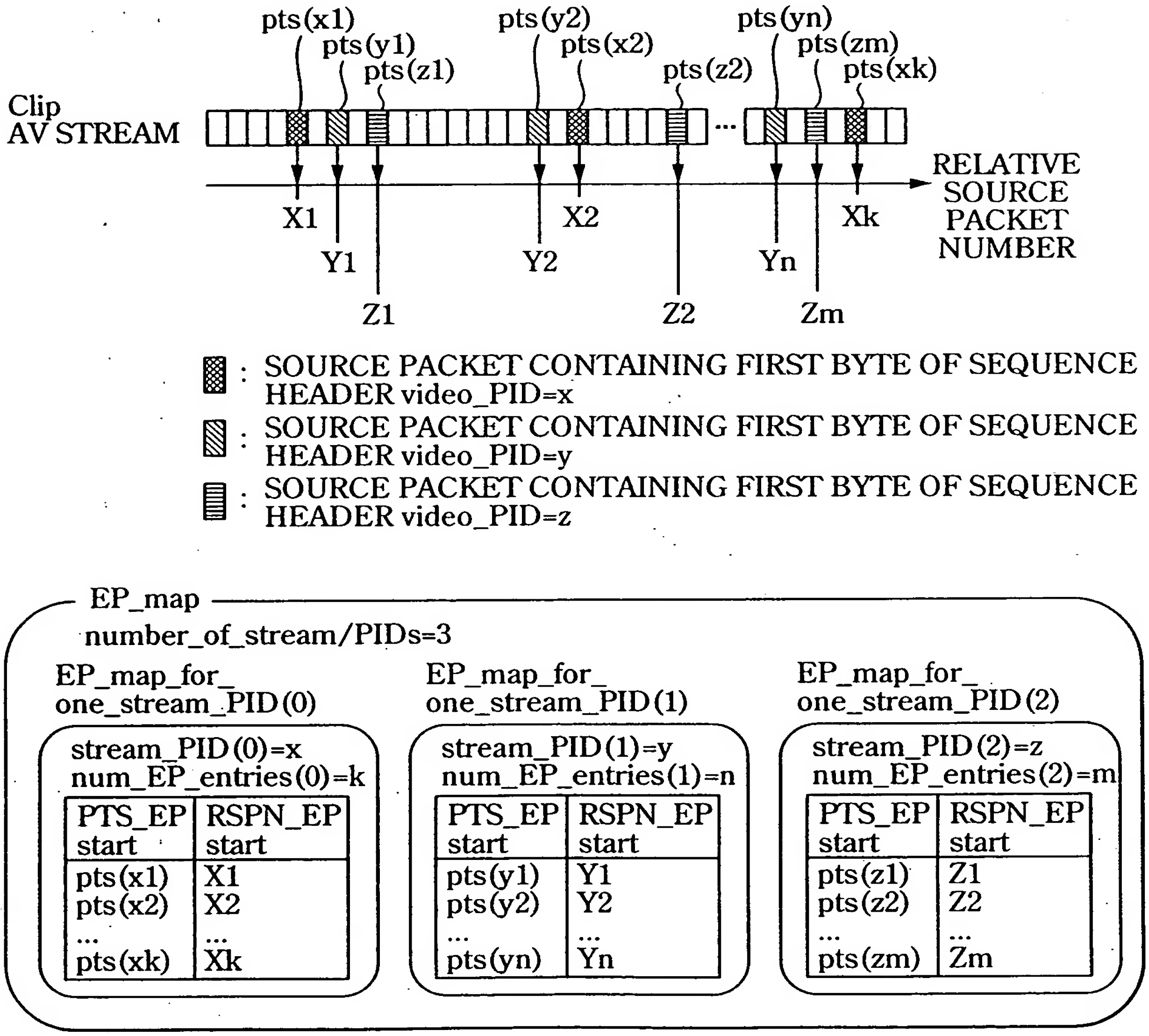
FIG.64

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| CPI(){ | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| reserved | 15 | bslbf |
| **CPI_type** | 1 | bslbf |
| if (CPI_type==0) | | |
| **EP_map()** | | |
| else | | |
| **TU_map()** | | |
| } | | |

FIG.65

| CPI_type | MEANING |
|---|---|
| 0 | EP map type |
| 1 | TU map type |

FIG.66



▩ : SOURCE PACKET CONTAINING FIRST BYTE OF SEQUENCE HEADER video_PID=x

▧ : SOURCE PACKET CONTAINING FIRST BYTE OF SEQUENCE HEADER video_PID=y

▤ : SOURCE PACKET CONTAINING FIRST BYTE OF SEQUENCE HEADER video_PID=z

EP_map

number_of_stream/PIDs=3

EP_map_for_
one_stream_PID(0)

stream_PID(0)=x
num_EP_entries(0)=k

| PTS_EP start | RSPN_EP start |
|---|---|
| pts(x1) | X1 |
| pts(x2) | X2 |
| ... | ... |
| pts(xk) | Xk |

EP_map_for_
one_stream_PID(1)

stream_PID(1)=y
num_EP_entries(1)=n

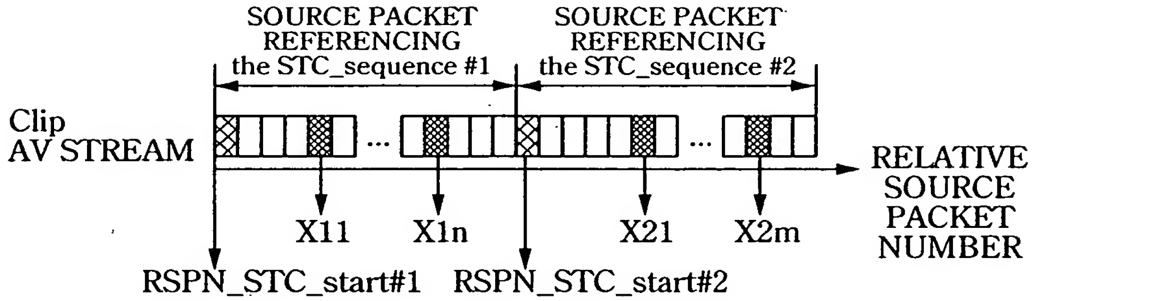
| PTS_EP start | RSPN_EP start |
|---|---|
| pts(y1) | Y1 |
| pts(y2) | Y2 |
| ... | ... |
| pts(yn) | Yn |

EP_map_for_
one_stream_PID(2)

stream_PID(2)=z
num_EP_entries(2)=m

| PTS_EP start | RSPN_EP start |
|---|---|
| pts(z1) | Z1 |
| pts(z2) | Z2 |
| ... | ... |
| pts(zm) | Zm |

FIG.67


SOURCE PACKET REFERENCING the STC_sequence #1
SOURCE PACKET REFERENCING the STC_sequence #2
Clip AV STREAM
RELATIVE SOURCE PACKET NUMBER
X11
X1n
X21
X2m
RSPN_STC_start#1
RSPN_STC_start#2

▩ : SOURCE PACKET CONTAINING FIRST BYTE OF SEQUENCE HEADER video_PID=x

☒ : SOURCE PACKET REFERENCED BY RSPN_STC_start (DEFINED IN the STC_into)

EP_map_for_one_steram_ PID
video_PID=x

| PTS_EP start | RSPN_EP start |
|---|---|
| pts(x11)<br>...<br>pts(x1n) | X11<br>...<br>X1n |
| pts(x21)<br>...<br>pts(x2m) | X21<br>...<br>X2m |

DATA BELONGING TO STC_sequence #1

boundary

DATA BELONGING TO STC_sequence #2

RSPN_STC_start #2<X21

FIG.68

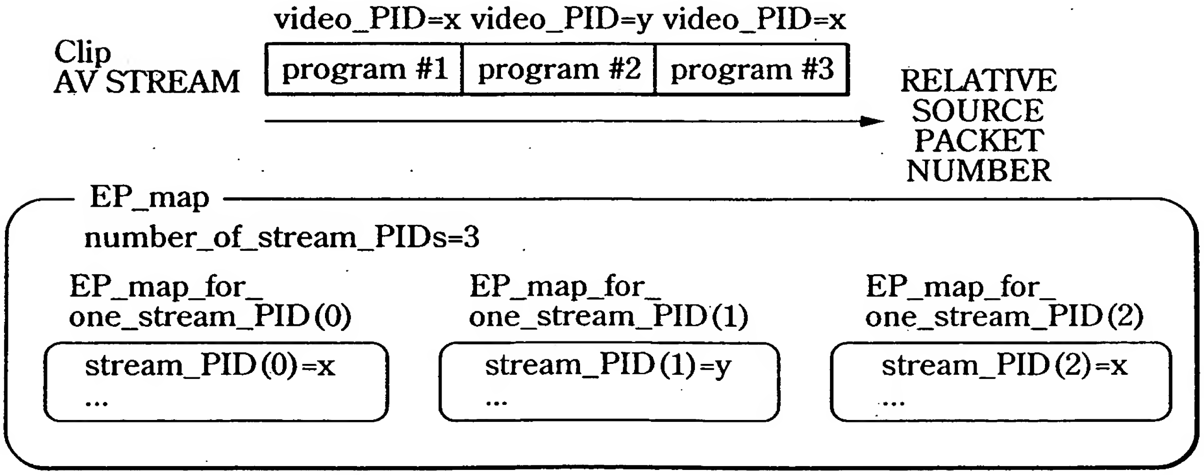

video_PID=x video_PID=y video_PID=x
Clip AV STREAM
program #1
program #2
program #3
RELATIVE SOURCE PACKET NUMBER
EP_map
number_of_stream_PIDs=3
EP_map_for_ one_stream_PID(0)
stream_PID(0)=x
...
EP_map_for_ one_stream_PID(1)
stream_PID(1)=y
...
EP_map_for_ one_stream_PID(2)
stream_PID(2)=x
...

FIG.69

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| EP_map(){ | | |
| reserved | 12 | bslbf |
| **EP_type** | 4 | uimsbf |
| **number_of_stream_PIDs** | 16 | uimsbf |
| for (k=0;k<*number_of_stream_PIDs*;k++){ | | |
| **stream_PID*(k)*** | 16 | bslbf |
| **num_EP_entries*(k)*** | 32 | uimsbf |
| **EP_map_for_one_stream_PID_Start_address*(k)*** | 32 | uimsbf |
| } | | |
| for (i=0;i<X;i++){ | | |
| **padding_word** | 16 | bslbf |
| } | | |
| for (k=0;k<*number_of_stream_PIDs*;k++){ | | |
| **EP_map_for_one_stream_PID**(*num_EP_entries(k)*) | | |
| for (i=0;i<Y;i++){ | | |
| **padding_word** | 16 | bslbf |
| } | | |
| } | | |
| } | | |

FIG.70

| EP_type | MEANING |
|---|---|
| 0 | video |
| 1 | audio |
| 2-15 | reserved |

FIG.71

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| EP_map_for_one_stream_PID(*N*) { | | |
| for (i=0;i<*N*;i++){ | | |
| **PTS_EP_start** | 32 | uimsbf |
| **RSPN_EP_start** | 32 | uimsbf |
| } | | |
| } | | |

FIG.72

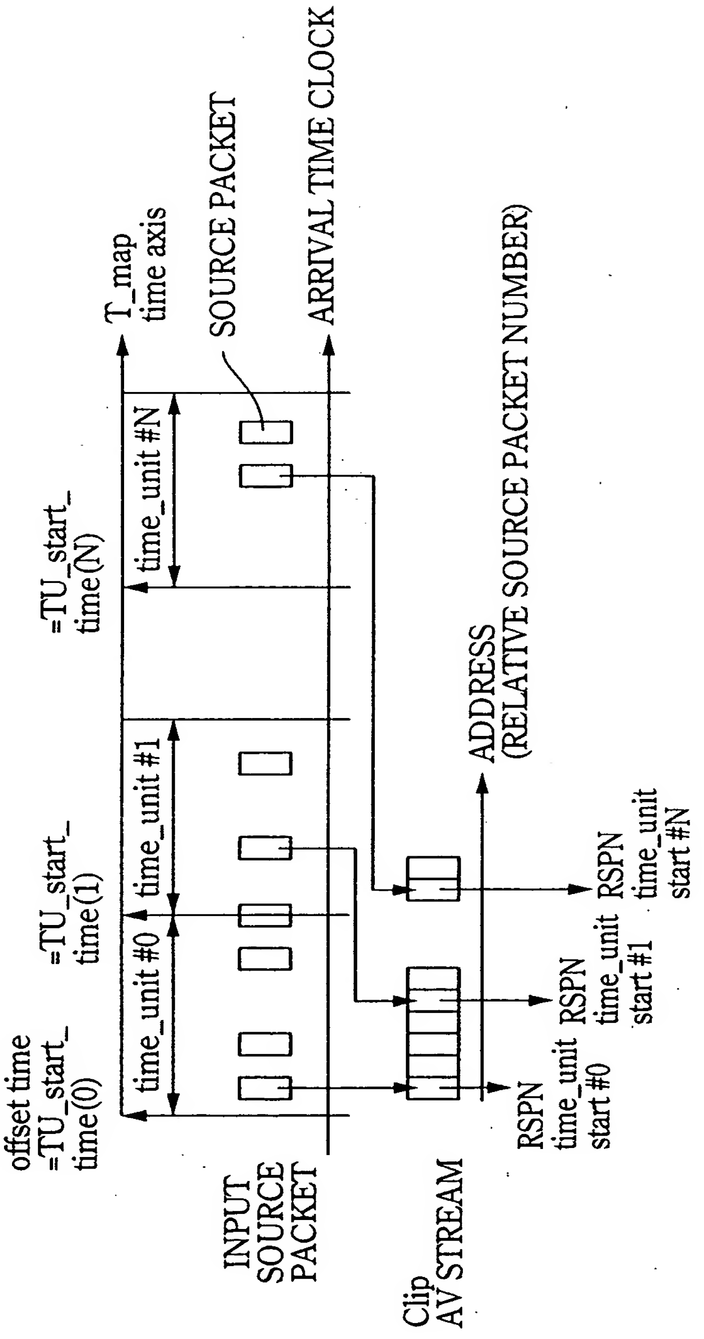

offset time
=TU_start_
time(0)
=TU_start_
time(1)
=TU_start_
time(N)
T_map
time axis
time_unit #0
time_unit #1
time_unit #N
SOURCE PACKET
ARRIVAL TIME CLOCK
INPUT
SOURCE
PACKET
Clip
AV STREAM
RSPN
time_unit
start #0
RSPN
time_unit
start #1
RSPN
time_unit
start #N
ADDRESS
(RELATIVE SOURCE PACKET NUMBER)

FIG.73

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| TU_map() { | | |
| **offset_time** | 32 | bslbf |
| **time_unit_size** | 32 | uimsbf |
| **number_of_time_unit_entries** | 32 | uimsbf |
| for (k=0;k<*number_of_time_unit_entries*;k++) | | |
| **RSPN_time_unit_start** | 32 | uimsbf |
| } | | |

FIG.74

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| ClipMark(){ | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| **number_of_Clip_marks** | 16 | uimsbf |
| for (i=0; i<*number_of_clip_marks*; i++){ | | |
| reserved | 8 | bslbf |
| **mark_type** | 8 | bslbf |
| **mark_time_stamp** | 32 | uimsbf |
| **STC_sequence_id** | 8 | uimsbf |
| reserved | 24 | bslbf |
| **character_set** | 8 | bslbf |
| **name_length** | 8 | uimsbf |
| **mark_name** | 8*256 | bslbf |
| **ref_thumbnail_index** | 16 | uimsbf |
| } | | |
| } | | |

FIG.75

| Mark_type | MEANING | COMMENT |
|---|---|---|
| 0x00-0x8F | reserved | Reserved for PlayListMark0 |
| 0x90 | Event-start mark | MARK POINT INDICATING PROGRAM START POINT |
| 0x91 | Local event-start mark | MARK POINT INDICATING LOCAL SCENE IN PROGRAM |
| 0x92 | Scene-start mark | MARK POINT SHOWING SCENE CHANGE POINT |
| 0x93-0xFF | reserved | |

FIG.76

| CPI_type<br>in the PlayList() | SEMANTICS OF mark_time_stamp |
|---|---|
| EP_map type | mark_time_stamp MUST INDICATE UPPER 32 BITS OF 33 BIT LENGTH PTS CORRESPONDING TO PRESENTATION UNIT REFERENCED BY MARK. |
| TU_map type | mark_time_stamp MUST BE TIME ON TU_map_time_axis AND MUST BE ROUNDED TO time_unit PRECISION. mark_time_stamp IS CALCULATED BY FOLLOWING EQUATION:<br>$\text{mark\_time\_stamp} = \text{TU\_start\_time} \,\%2^{32}$ |

FIG.77

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| ClipMark(){ | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| **number_of_Clip_marks** | 16 | uimsbf |
| for (i=0; i<*number_of_Clip_marks*; i++){ | | |
| reserved | 8 | bslbf |
| **mark_type** | 8 | bslbf |
| **reserved_for_MakerID** | 16 | bslbf |
| *mark_entry()* | | |
| *representative_picture_entry()* | | |
| **ref_thumbnail_index** | 16 | uimsbf |
| } | | |
| } | | |

# FIG.78

| Mark_type | MEANING | COMMENT |
|---|---|---|
| 0x00-0x8F | reserved | Reserved for PlayListMark() |
| 0x90 | Event-start mark | MARK POINT INDICATING PROGRAM START POINT |
| 0x91 | Local event-start mark | MARK POINT INDICATING LOCAL SCENE IN PROGRAM |
| 0x92 | Scene-start mark | MARK POINT INDICATING SCENE START POINT |
| 0x93 | Scene-end mark | MARK POINT INDICATING SCENE END POINT |
| 0x94 | CM-start mark | MARK POINT INDICATING CM START POINT |
| 0x95 | CM-end mark | MARK POINT INDICATING CM END POINT |
| 0x96-0xBF | DVR FORMAT IS RESERVED FOR FUTURE EXTENSION OF ClipMark | |
| 0xC0-0xFF | ALLOCATBLE TO MARK USED IN MAKER-UNIQUE APPLICCATION | |

# FIG.79

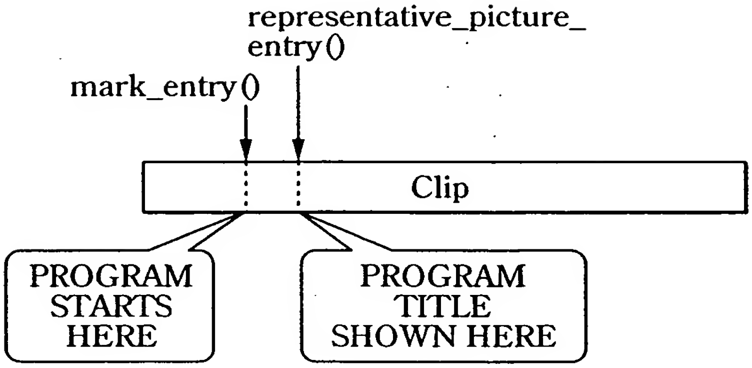


FIG.80

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| **mark_entry()/representative_picture_entry(){** | | |
| **mark_time_stamp** | 32 | uimsbf |
| **STC_sequence_id** | 8 | uimsbf |
| reserved | 24 | bslbf |
| } | | |

FIG.81

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| mark_entry()/representative_picture_entry(){ | | |
| **RSPN_ref_EP_start** | 32 | uimsbf |
| **offset_num_pictures** | 32 | uimsbf |
| } | | |

FIG.82

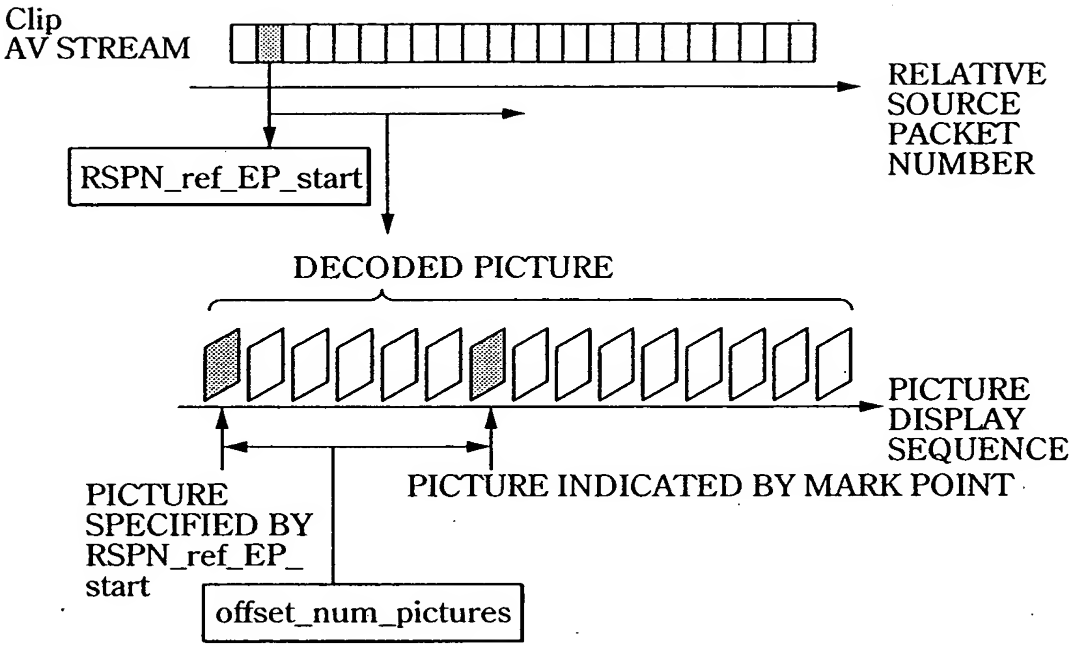
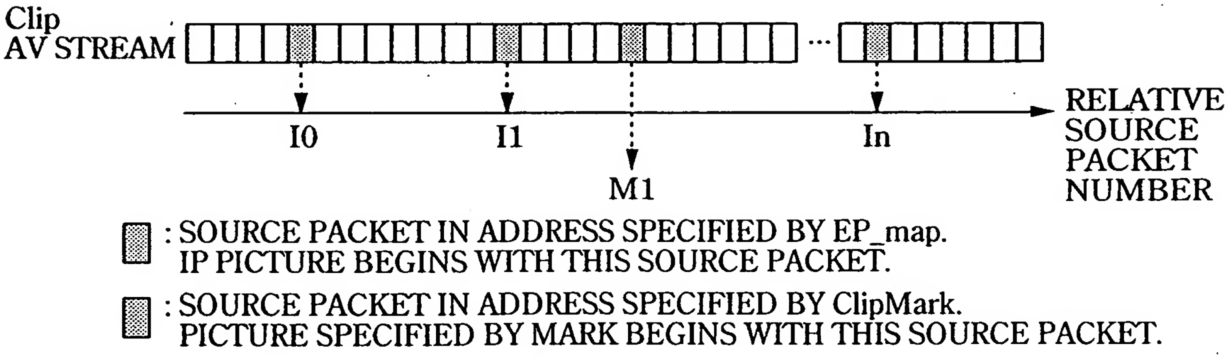

Clip
AV STREAM
RELATIVE SOURCE PACKET NUMBER
RSPN_ref_EP_start
DECODED PICTURE
PICTURE DISPLAY SEQUENCE
PICTURE INDICATED BY MARK POINT
PICTURE SPECIFIED BY RSPN_ref_EP_ start
offset_num_pictures

FIG.83

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| mark_entry()/representative_picture_entry(){ | | |
| **RSPN_mark_point** | 32 | uimsbf |
| } | | |

FIG.84



▨ : SOURCE PACKET IN ADDRESS SPECIFIED BY EP_map.
IP PICTURE BEGINS WITH THIS SOURCE PACKET.

▨ : SOURCE PACKET IN ADDRESS SPECIFIED BY ClipMark.
PICTURE SPECIFIED BY MARK BEGINS WITH THIS SOURCE PACKET.

FIG.85

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| menu.thmb/mark.thmb() { | | |
| reserved | 256 | bslbf |
| **Thumbnail()** | | |
| for (i=0;i<N1;i++) | | |
| **padding_word** | 16 | bslbf |
| } | | |

FIG.86

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| Thumbnail() { | | |
| **version_number** | 8*4 | char |
| **length** | 32 | uimsbf |
| if (length !=0) { | | |
| **tn_blocks_start_address** | 32 | bslbf |
| **number_of_thumbnails** | 16 | uimsbf |
| **tn_block_size** | 16 | uimsbf |
| **number_of_tn_blocks** | 16 | uimsbf |
| reserved | 16 | bslbf |
| for (i=0; i<*number_of_thumbnails*; i++) { | | |
| **thumbnail_index** | 16 | uimsbf |
| **thumbnail_picture_format** | 8 | bslbf |
| reserved | 8 | bslbf |
| **picture_data_size** | 32 | uimsbf |
| **start_tn_block_number** | 16 | uimsbf |
| **x_picture_length** | 16 | uimsbf |
| **y_picture_length** | 16 | uimsbf |
| reserved | 16 | uimsbf |
| } | | |
| **stuffing_bytes** | 8*2*L1 | bslbf |
| for(k=0; k<*number_of_tn_blocks*; k++) { | | |
| **tn_block** | tn_block_ size*1024*8 | |
| } | | |
| } | | |
| } | | |

FIG.87

| Thumbnail_picture_format | MEANING |
|---|---|
| 0x00 | MPEG-2 Video I-picture |
| 0x01 | DCF (restricted JPEG) |
| 0x02 | PNG |
| 0x03-0xff | reserved |

FIG.88

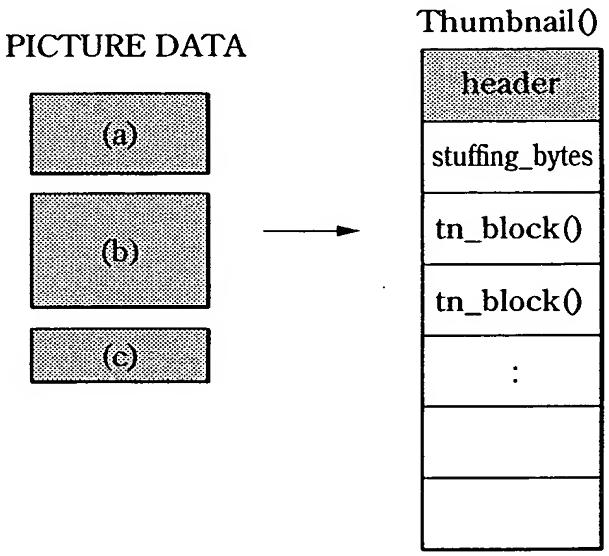


FIG.89A

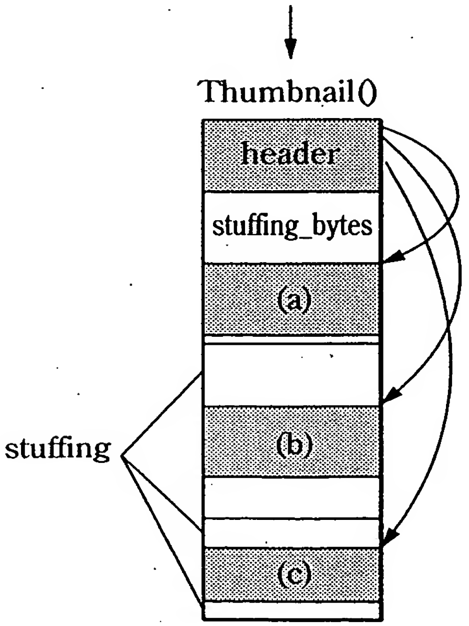


FIG.89B

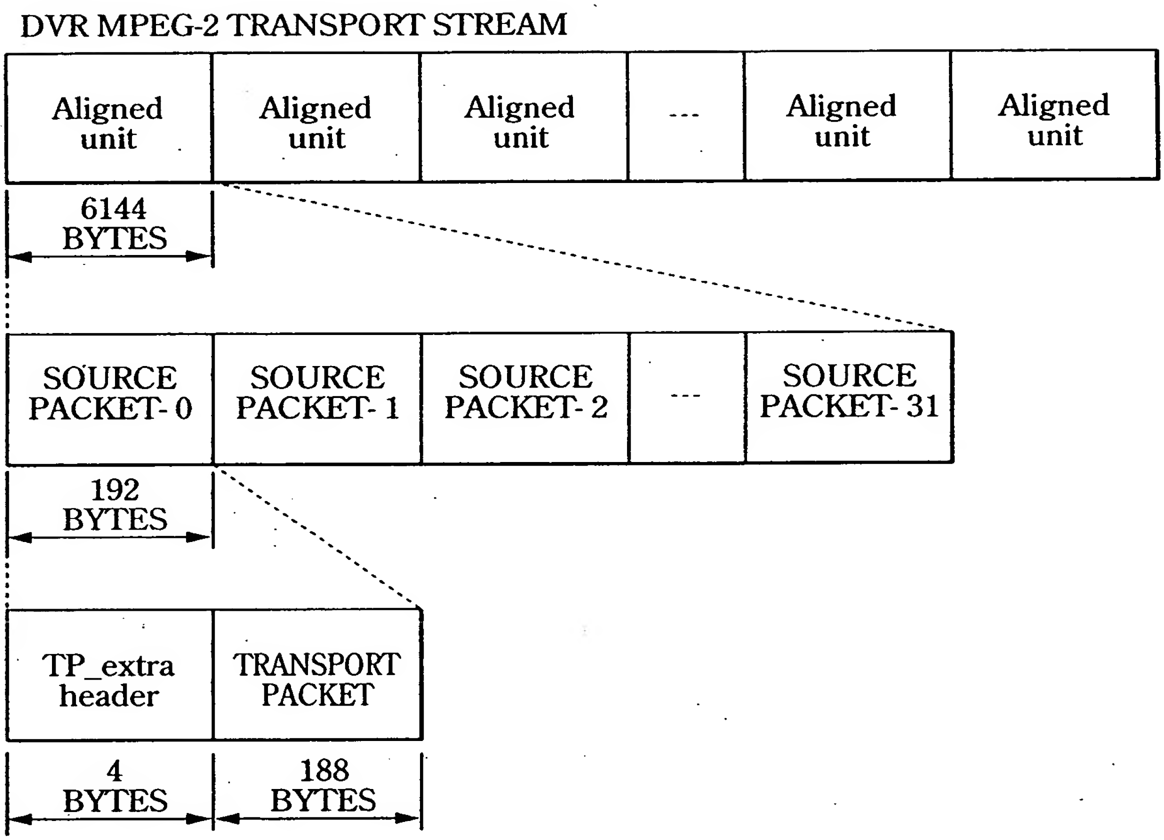

DVR MPEG-2 TRANSPORT STREAM
Aligned unit
Aligned unit
Aligned unit
---
Aligned unit
Aligned unit
6144 BYTES
SOURCE PACKET- 0
SOURCE PACKET- 1
SOURCE PACKET- 2
---
SOURCE PACKET- 31
192 BYTES
TP_extra header
TRANSPORT PACKET
4 BYTES
188 BYTES

FIG.90

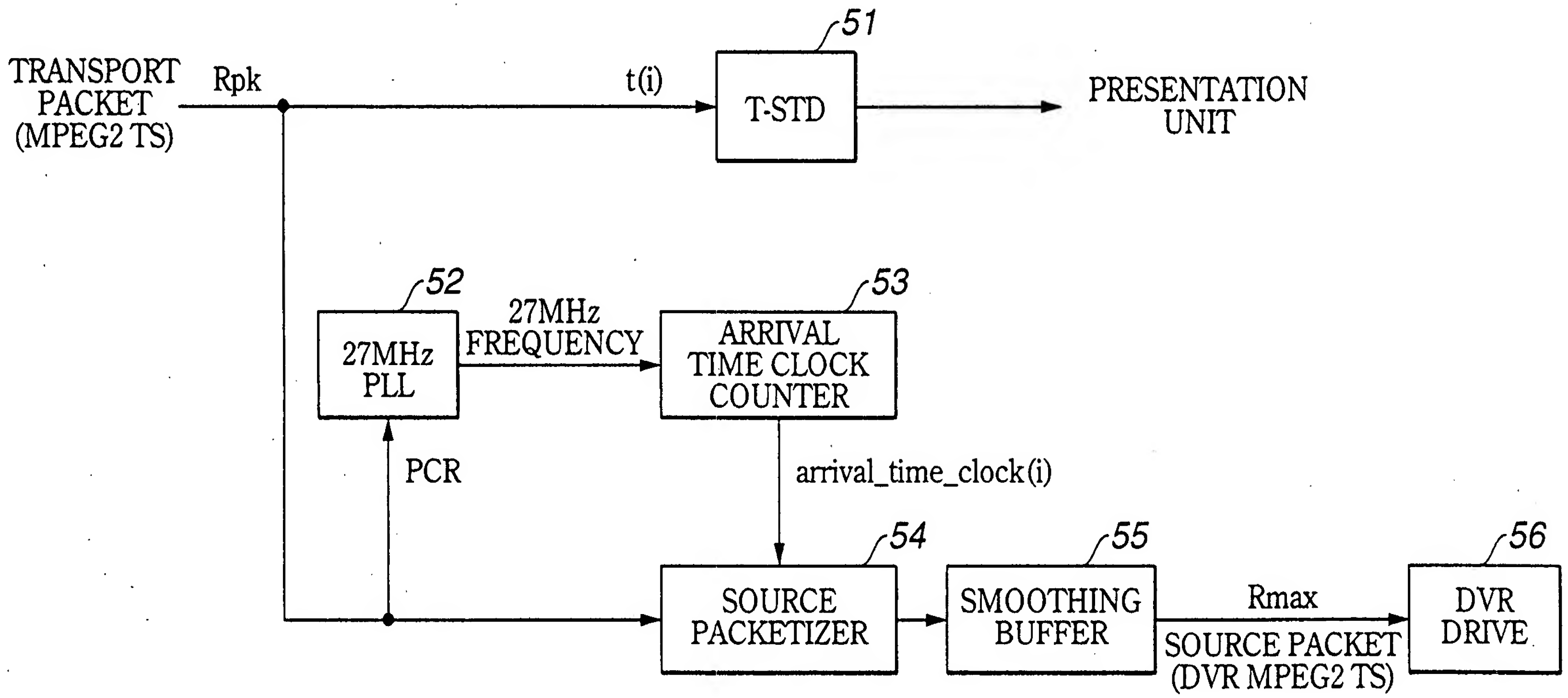

TRANSPORT PACKET (MPEG2 TS)
Rpk
t(i)
51
T-STD
PRESENTATION UNIT
52
27MHz PLL
27MHz FREQUENCY
53
ARRIVAL TIME CLOCK COUNTER
PCR
arrival_time_clock(i)
54
SOURCE PACKETIZER
55
SMOOTHING BUFFER
Rmax
SOURCE PACKET (DVR MPEG2 TS)
56
DVR DRIVE

FIG.91

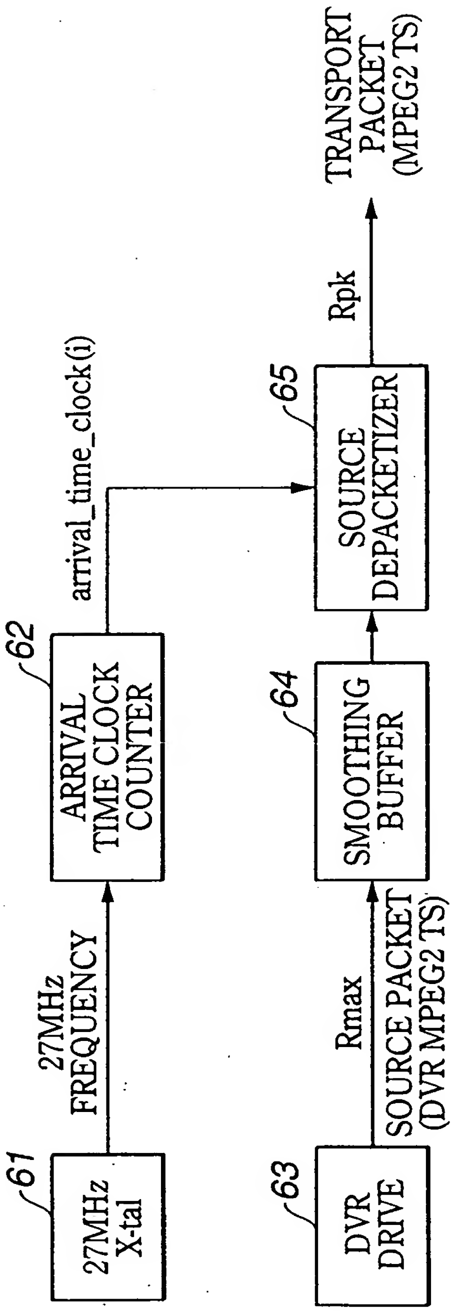

61
27MHz X-tal
27MHz FREQUENCY
62
ARRIVAL TIME CLOCK COUNTER
arrival_time_clock(i)
63
DVR DRIVE
Rmax
SOURCE PACKET (DVR MPEG2 TS)
64
SMOOTHING BUFFER
65
SOURCE DEPACKETIZER
Rpk
TRANSPORT PACKET (MPEG2 TS)

FIG.92

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| source_packet() { | | |
| **TP_extra_header()** | | |
| **trasport_packet()** | | |
| } | | |

FIG.93

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| TP_extra_header() { | | |
| **copy_permission_indicator** | 2 | uimsbf |
| **arrival_time_stamp** | 30 | uimsbf |
| } | | |

FIG.94

| copy_permission _indicator | MEANING |
|---|---|
| 00 | copy free |
| 01 | no more copy |
| 10 | copy once |
| 11 | copy prohibited |

FIG.95

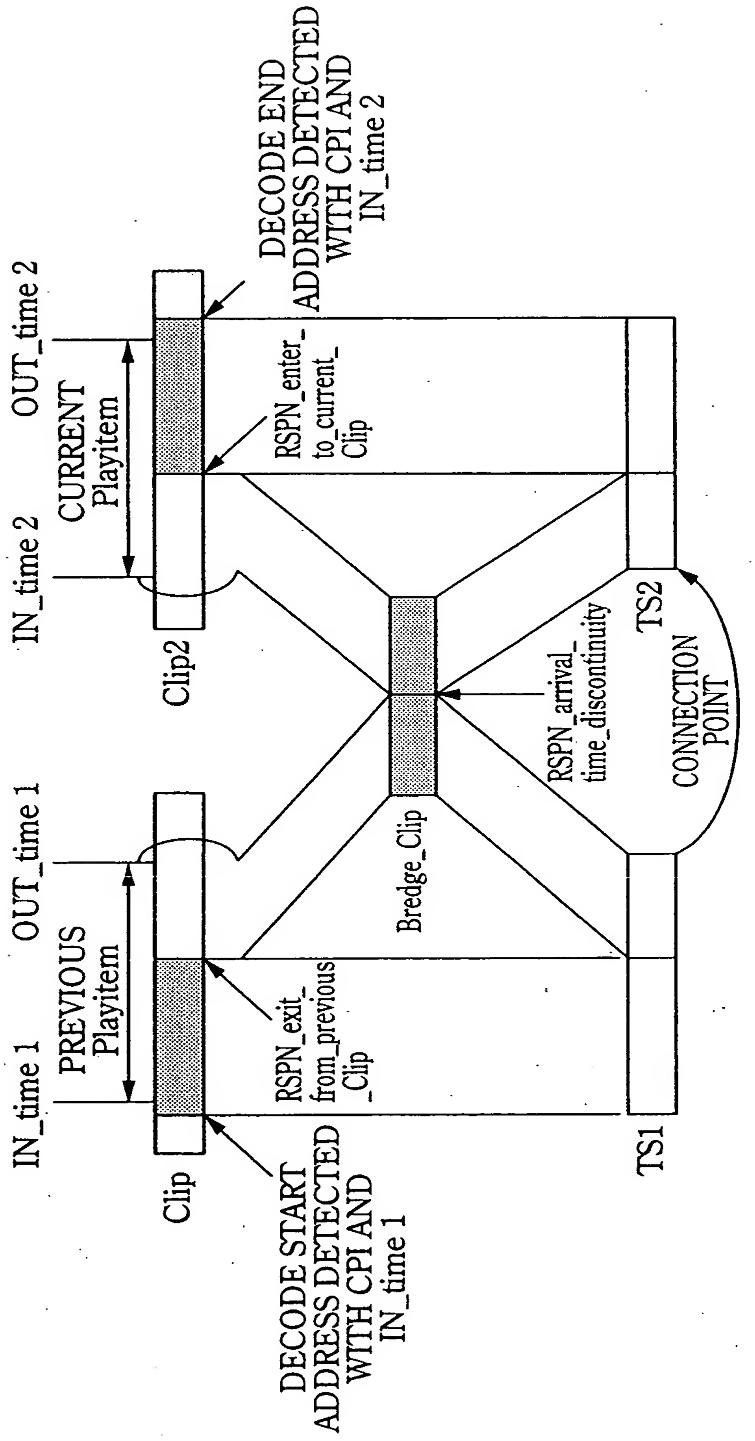

IN_time 1
OUT_time 1
PREVIOUS Playitem
Clip
DECODE START ADDRESS DETECTED WITH CPI AND IN_time 1
RSPN_exit_ from_previous _Clip
TS1
Bredge_Clip
RSPN_arrival_ time_discontinuity
CONNECTION POINT
TS2
IN_time 2
OUT_time 2
CURRENT Playitem
Clip2
RSPN_enter_ to_current_ Clip
DECODE END ADDRESS DETECTED WITH CPI AND IN_time 2

FIG.96

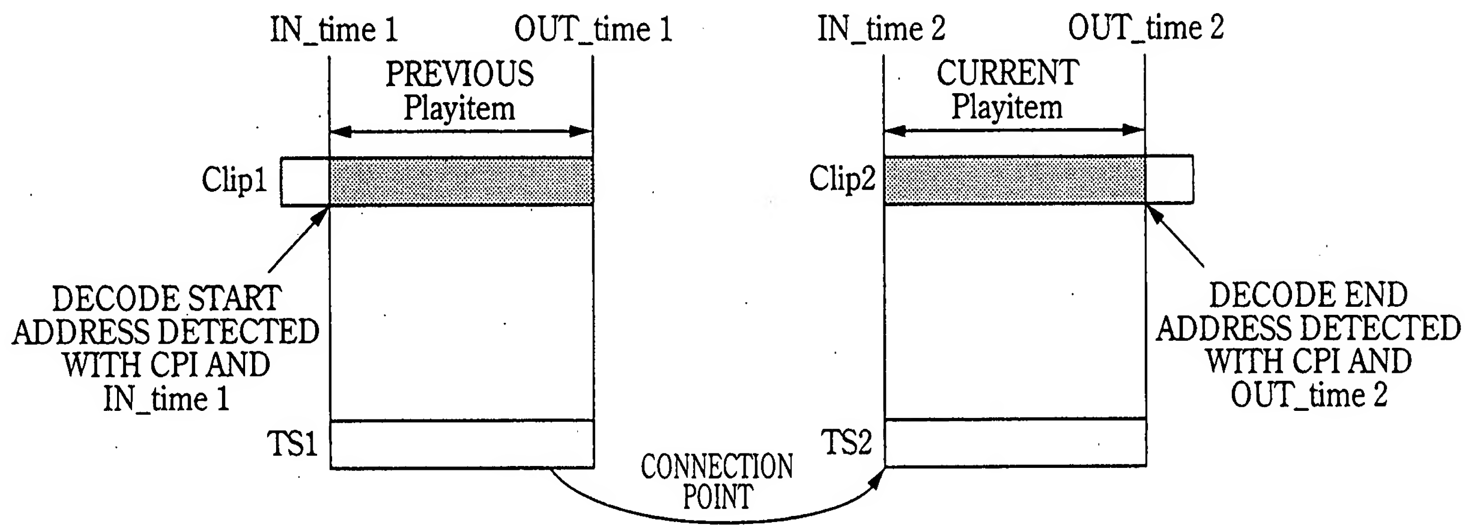

IN_time 1
OUT_time 1
IN_time 2
OUT_time 2
PREVIOUS Playitem
CURRENT Playitem
Clip1
Clip2
DECODE START ADDRESS DETECTED WITH CPI AND IN_time 1
DECODE END ADDRESS DETECTED WITH CPI AND OUT_time 2
TS1
TS2
CONNECTION POINT

FIG.97

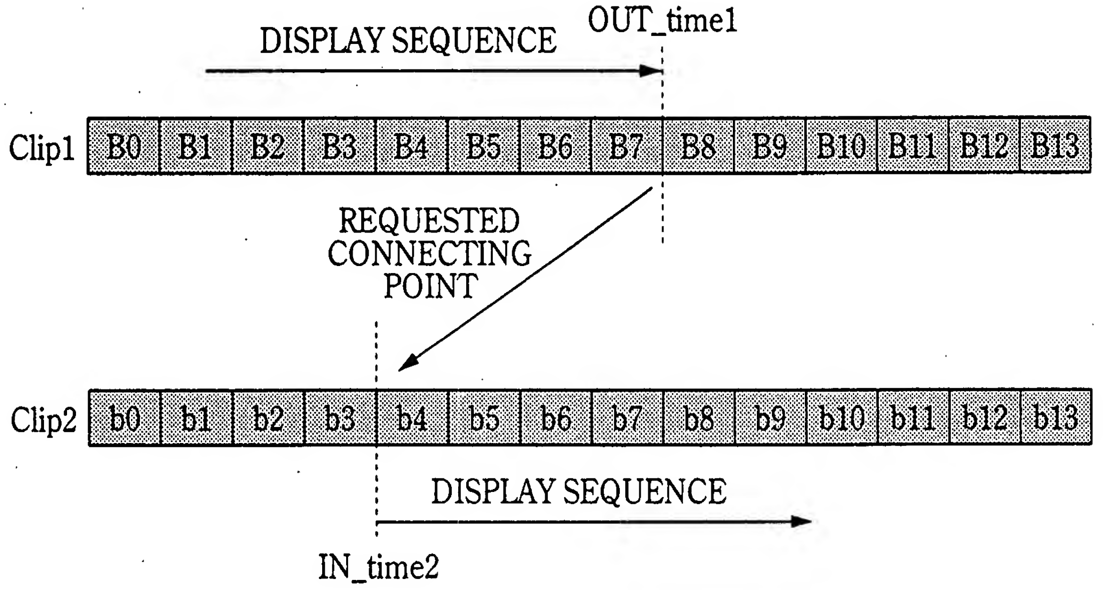

DISPLAY SEQUENCE
OUT_time1
Clip1
B0
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
REQUESTED CONNECTING POINT
Clip2
b0
b1
b2
b3
b4
b5
b6
b7
b8
b9
b10
b11
b12
b13
DISPLAY SEQUENCE
IN_time2

FIG.98

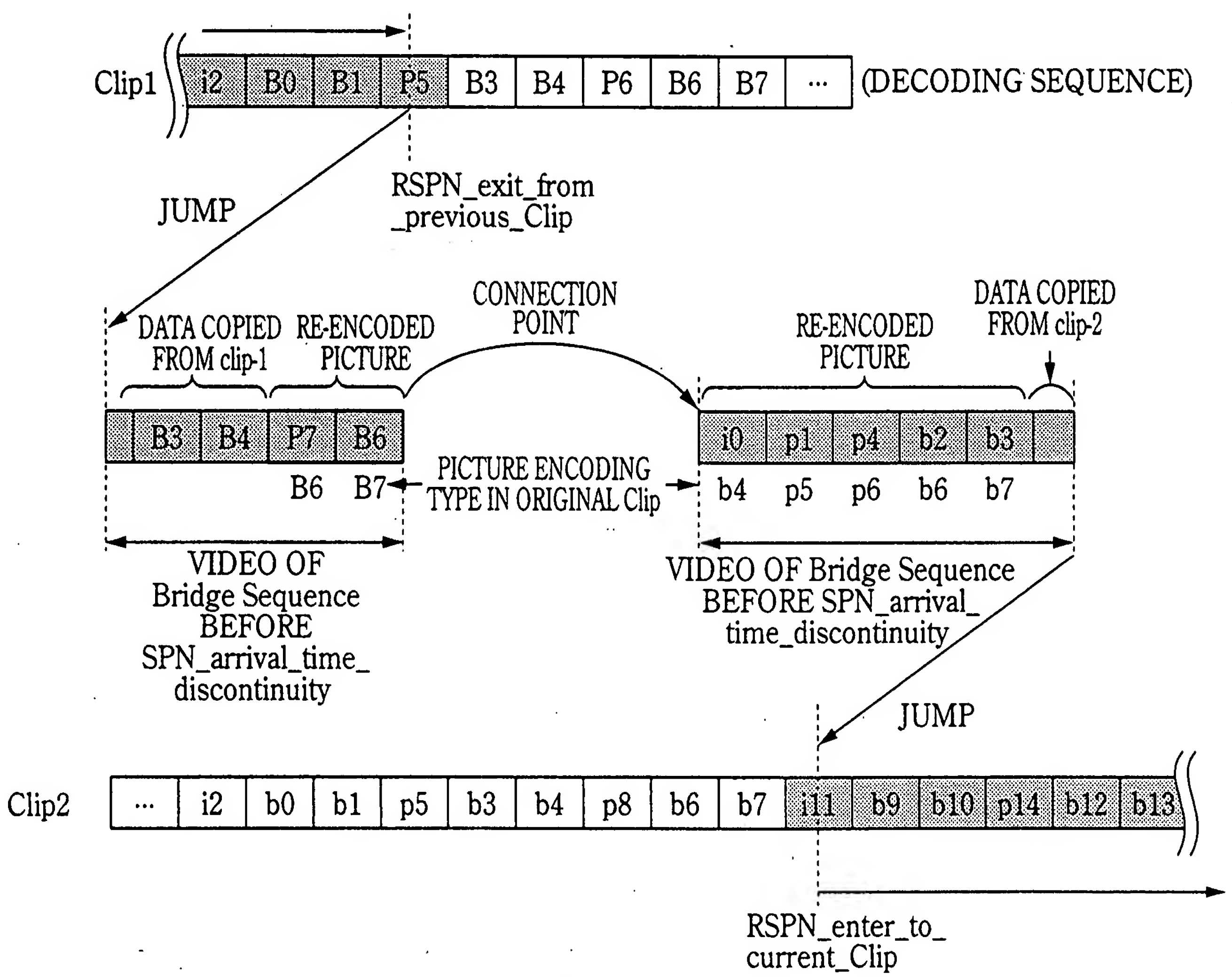

Clip1
i2
B0
B1
P5
B3
B4
P6
B6
B7
...
(DECODING SEQUENCE)
JUMP
RSPN_exit_from
_previous_Clip
DATA COPIED
FROM clip-1
RE-ENCODED
PICTURE
CONNECTION
POINT
RE-ENCODED
PICTURE
DATA COPIED
FROM clip-2
B3
B4
P7
B6
i0
p1
p4
b2
b3
B6
B7
PICTURE ENCODING
TYPE IN ORIGINAL Clip
b4
p5
p6
b6
b7
VIDEO OF
Bridge Sequence
BEFORE
SPN_arrival_time_
discontinuity
VIDEO OF Bridge Sequence
BEFORE SPN_arrival_
time_discontinuity
JUMP
Clip2
...
i2
b0
b1
p5
b3
b4
p8
b6
b7
i11
b9
b10
p14
b12
b13
RSPN_enter_to_
current_Clip

FIG.99

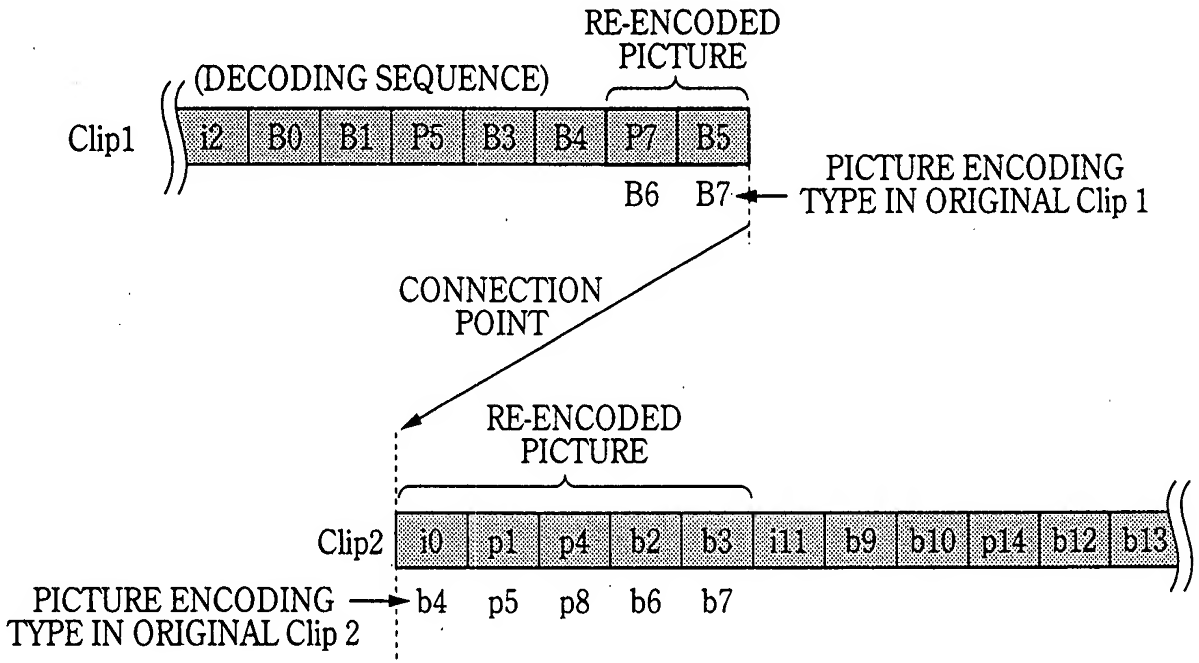

(DECODING SEQUENCE)
RE-ENCODED PICTURE
Clip1
i2
B0
B1
P5
B3
B4
P7
B5
B6
B7
PICTURE ENCODING TYPE IN ORIGINAL Clip 1
CONNECTION POINT
RE-ENCODED PICTURE
Clip2
i0
p1
p4
b2
b3
i11
b9
b10
p14
b12
b13
PICTURE ENCODING TYPE IN ORIGINAL Clip 2
b4
p5
p8
b6
b7

FIG.100

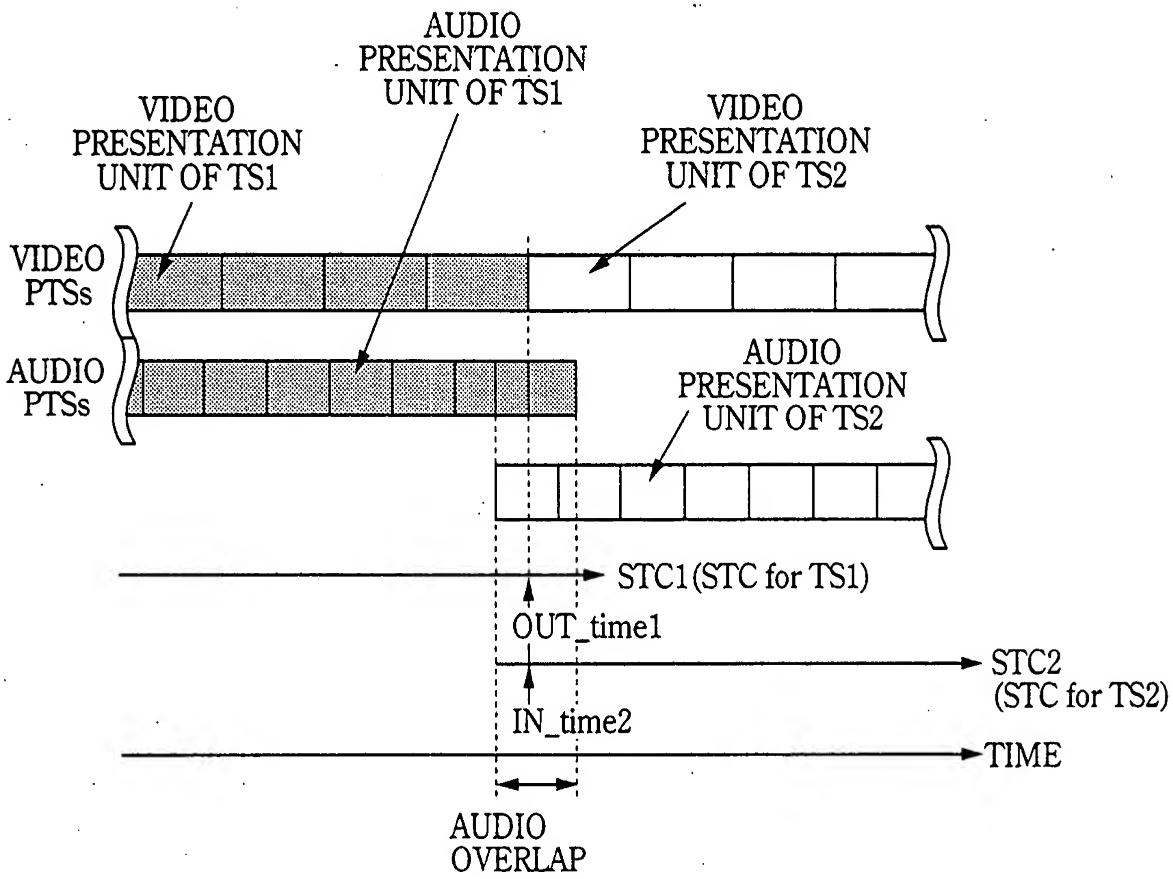

AUDIO
PRESENTATION
UNIT OF TS1
VIDEO
PRESENTATION
UNIT OF TS1
VIDEO
PRESENTATION
UNIT OF TS2
VIDEO
PTSs
AUDIO
PTSs
AUDIO
PRESENTATION
UNIT OF TS2
STC1 (STC for TS1)
OUT_time1
STC2
(STC for TS2)
IN_time2
TIME
AUDIO
OVERLAP

FIG.101

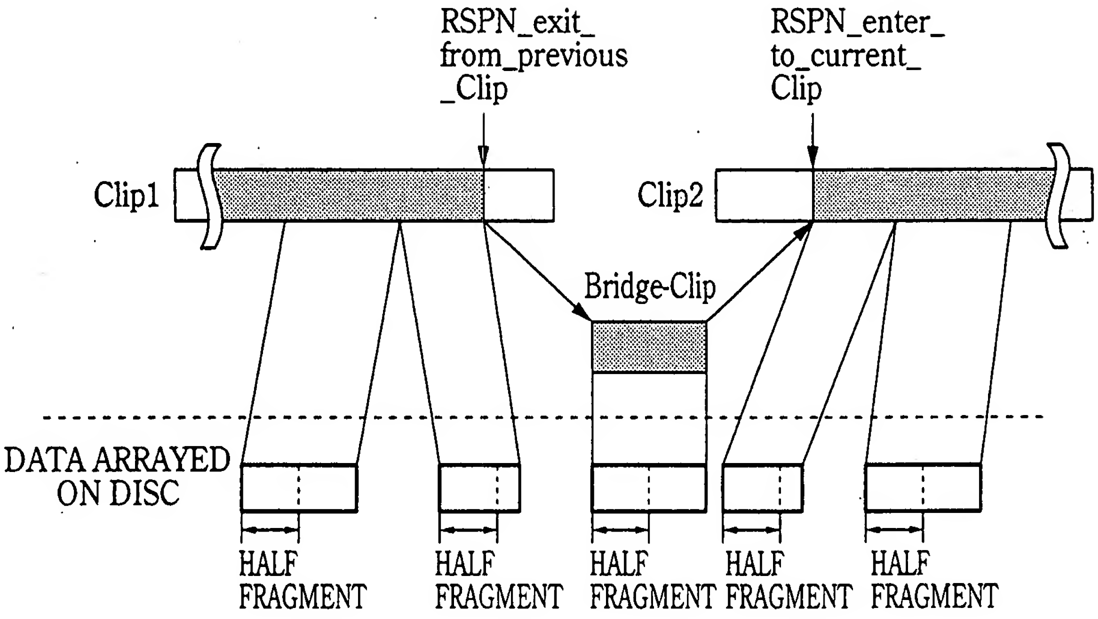

RSPN_exit_
from_previous
_Clip
RSPN_enter_
to_current_
Clip
Clip1
Clip2
Bridge-Clip
DATA ARRAYED
ON DISC
HALF
FRAGMENT
HALF
FRAGMENT
HALF
FRAGMENT
HALF
FRAGMENT
HALF
FRAGMENT

FIG.102

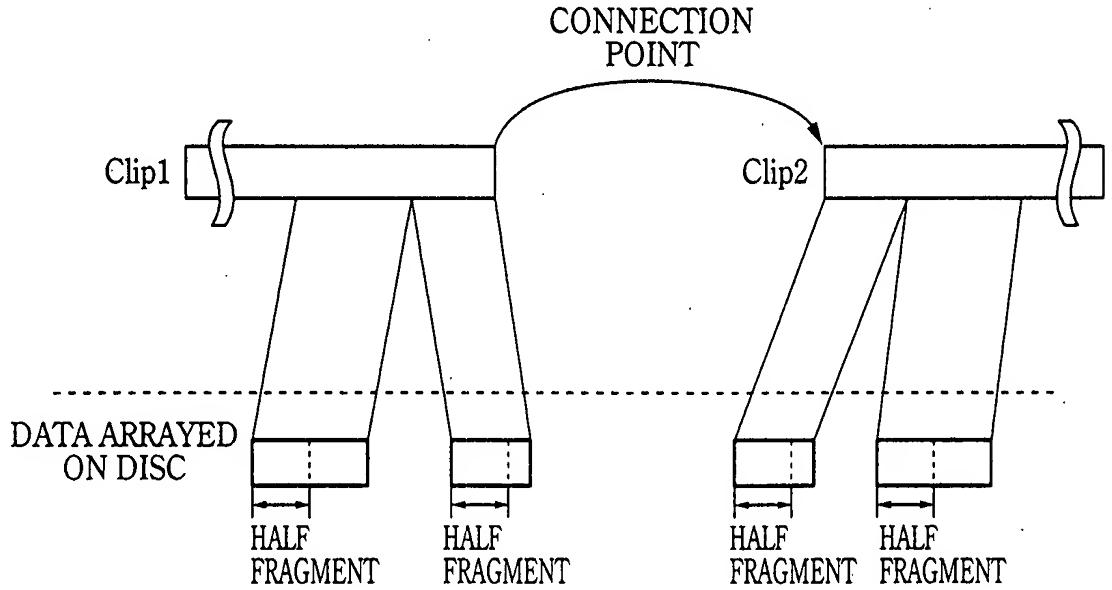

CONNECTION POINT
Clip1
Clip2
DATA ARRAYED ON DISC
HALF FRAGMENT
HALF FRAGMENT
HALF FRAGMENT
HALF FRAGMENT

FIG.103

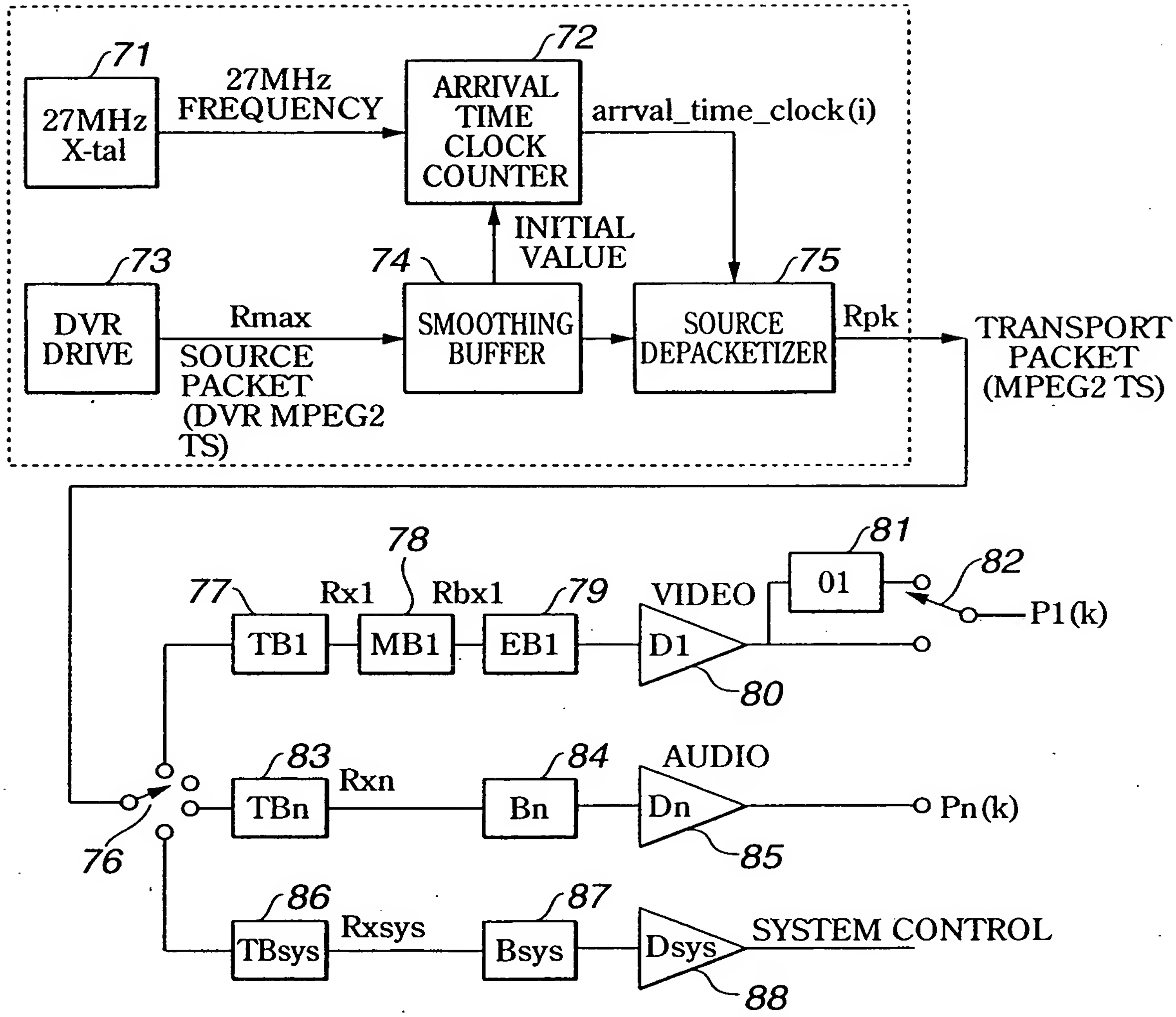

71
27MHz X-tal
27MHz FREQUENCY
72
ARRIVAL TIME CLOCK COUNTER
arrval_time_clock (i)
INITIAL VALUE
73
DVR DRIVE
Rmax
SOURCE PACKET (DVR MPEG2 TS)
74
SMOOTHING BUFFER
75
SOURCE DEPACKETIZER
Rpk
TRANSPORT PACKET (MPEG2 TS)
77
TB1
Rx1
78
MB1
Rbx1
79
EB1
VIDEO
D1
80
81
01
82
P1 (k)
83
TBn
Rxn
84
Bn
AUDIO
Dn
85
Pn (k)
76
86
TBsys
Rxsys
87
Bsys
Dsys
SYSTEM CONTROL
88

FIG.104

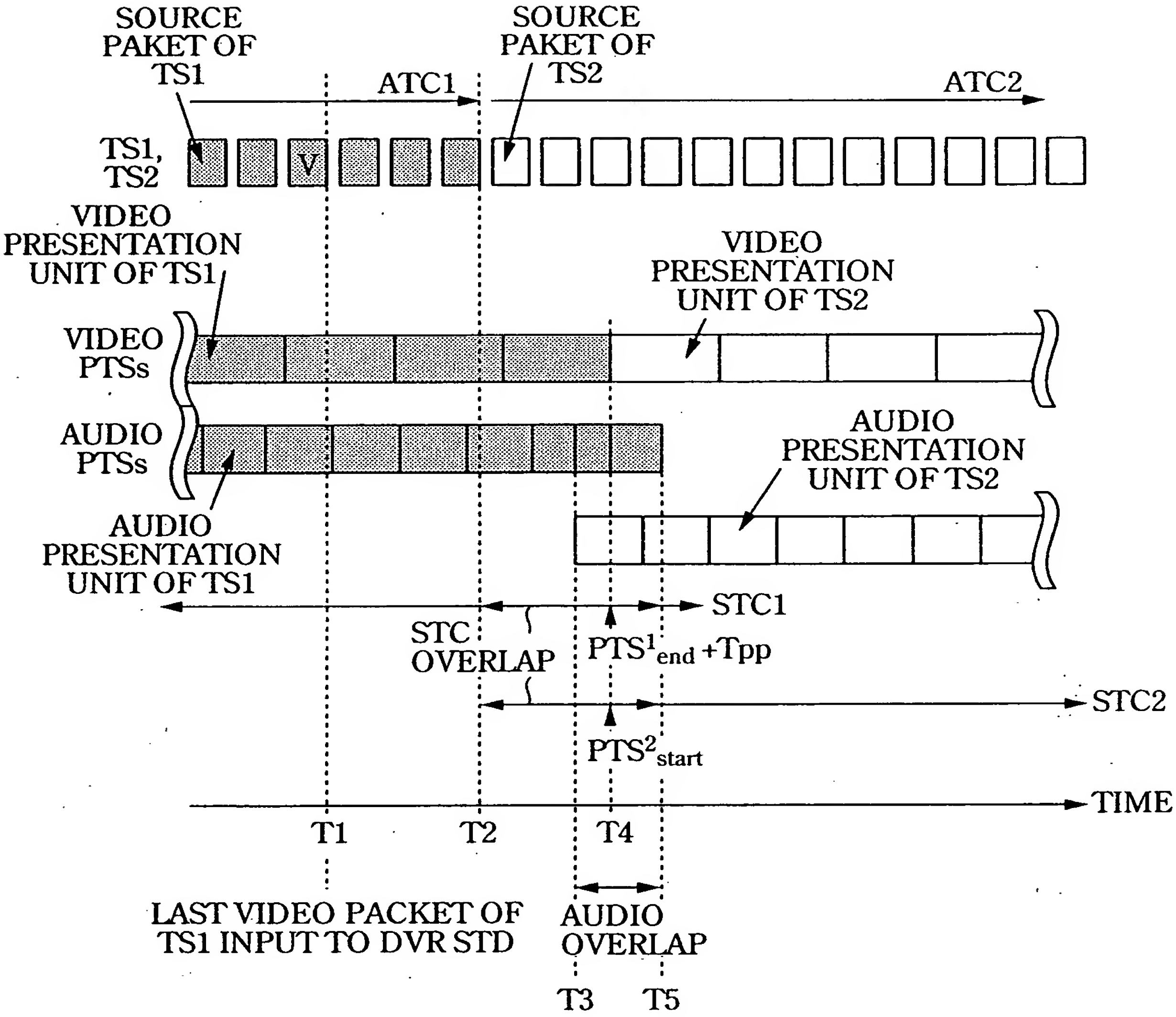

SOURCE PAKET OF TS1
SOURCE PAKET OF TS2
ATC1
ATC2
TS1, TS2
V
VIDEO PRESENTATION UNIT OF TS1
VIDEO PRESENTATION UNIT OF TS2
VIDEO PTSs
AUDIO PTSs
AUDIO PRESENTATION UNIT OF TS2
AUDIO PRESENTATION UNIT OF TS1
STC1
STC OVERLAP
PTS$^1_{end}$ +Tpp
STC2
PTS$^2_{start}$
TIME
T1
T2
T4
LAST VIDEO PACKET OF TS1 INPUT TO DVR STD
AUDIO OVERLAP
T3
T5

FIG.105

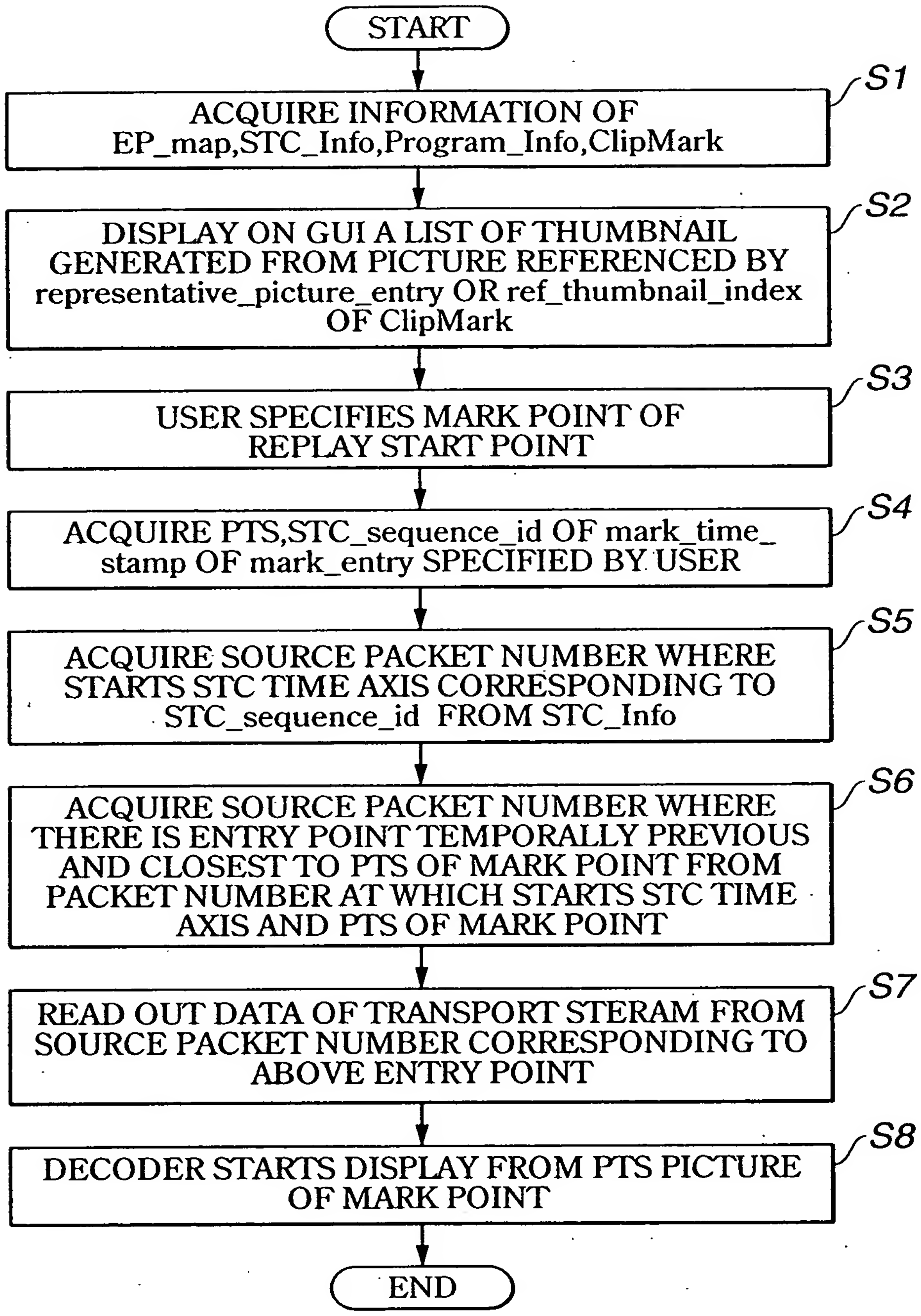

START
S1
ACQUIRE INFORMATION OF EP_map,STC_Info,Program_Info,ClipMark
S2
DISPLAY ON GUI A LIST OF THUMBNAIL GENERATED FROM PICTURE REFERENCED BY representative_picture_entry OR ref_thumbnail_index OF ClipMark
S3
USER SPECIFIES MARK POINT OF REPLAY START POINT
S4
ACQUIRE PTS,STC_sequence_id OF mark_time_stamp OF mark_entry SPECIFIED BY USER
S5
ACQUIRE SOURCE PACKET NUMBER WHERE STARTS STC TIME AXIS CORRESPONDING TO STC_sequence_id FROM STC_Info
S6
ACQUIRE SOURCE PACKET NUMBER WHERE THERE IS ENTRY POINT TEMPORALLY PREVIOUS AND CLOSEST TO PTS OF MARK POINT FROM PACKET NUMBER AT WHICH STARTS STC TIME AXIS AND PTS OF MARK POINT
S7
READ OUT DATA OF TRANSPORT STERAM FROM SOURCE PACKET NUMBER CORRESPONDING TO ABOVE ENTRY POINT
S8
DECODER STARTS DISPLAY FROM PTS PICTURE OF MARK POINT
END

FIG.106

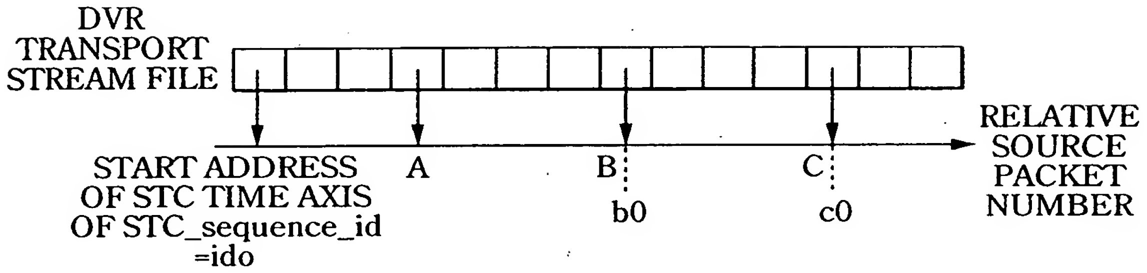

DVR TRANSPORT STREAM FILE
START ADDRESS OF STC TIME AXIS OF STC_sequence_id =ido
A
B
b0
C
c0
RELATIVE SOURCE PACKET NUMBER

FIG.107

EP_map

| RSPN_EP_ start | PTS_EP_ start |
|---|---|
| . . . | . . . |
| A | PTS(A) |
| B | PTS(B) |
| C | PTS(C) |
| . . . | . . . |

FIG.108

ClipMark

| Mark_type | mark_entry | | representative _ picture _entry | |
|---|---|---|---|---|
| | Mark_ Time_stamp | STC_sequence_ id | Mark_ Time_stamp | STC_sequence_ id |
| . . . | . . . | . . . | | |
| 0x92(scene start) | PTS(a1) | id0 | PTS(a2) | id0 |
| 0x94(CM start) | PTS(b0) | id0 | PTS(b0) | id0 |
| 0x95(CM end) | PTS(c0) | id0 | PTS(c0) | id0 |
| . . . | . . . | . . . | . . . | . . . |

FIG.109

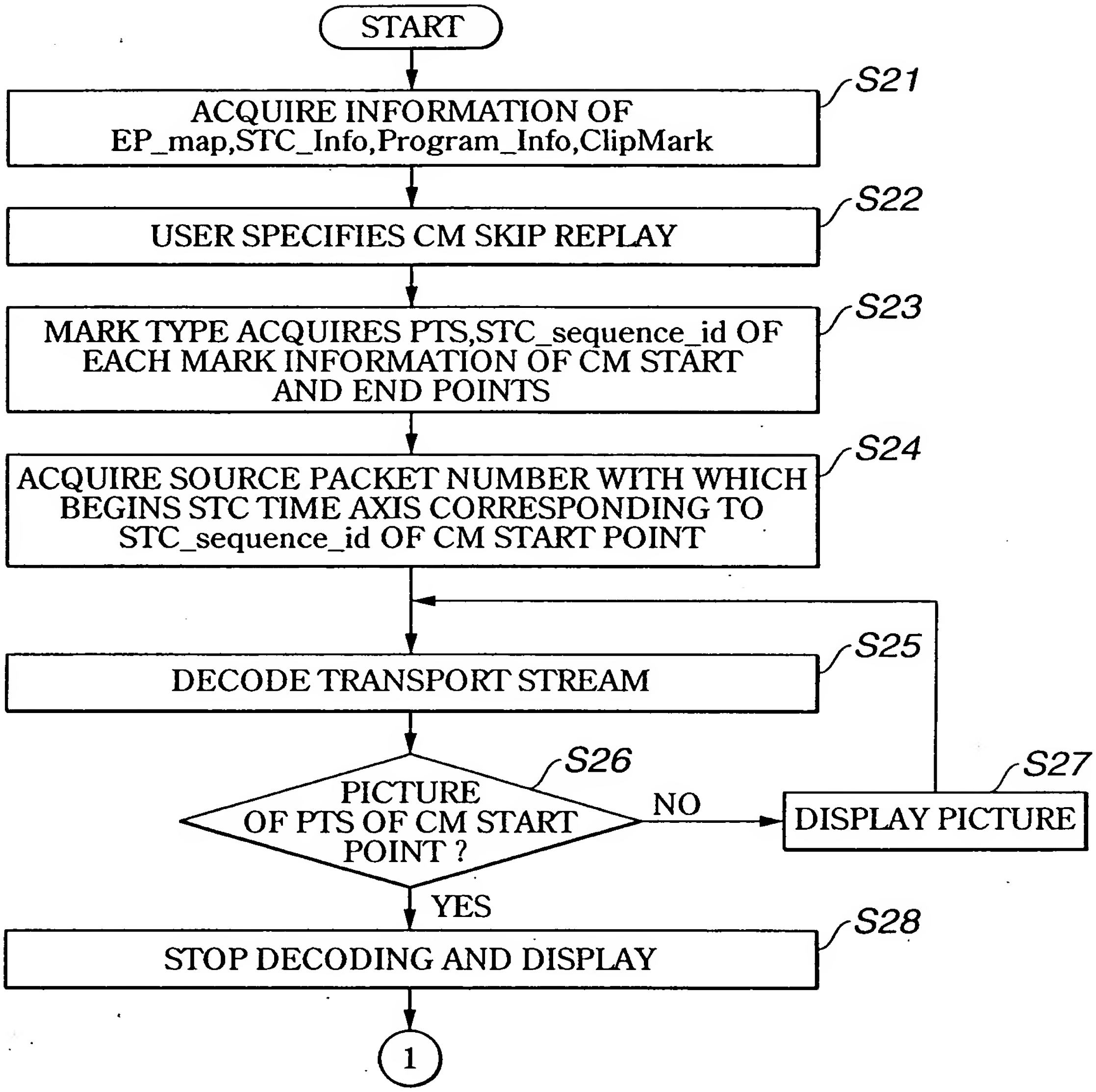

START
ACQUIRE INFORMATION OF EP_map,STC_Info,Program_Info,ClipMark
S21
USER SPECIFIES CM SKIP REPLAY
S22
MARK TYPE ACQUIRES PTS,STC_sequence_id OF EACH MARK INFORMATION OF CM START AND END POINTS
S23
ACQUIRE SOURCE PACKET NUMBER WITH WHICH BEGINS STC TIME AXIS CORRESPONDING TO STC_sequence_id OF CM START POINT
S24
DECODE TRANSPORT STREAM
S25
PICTURE OF PTS OF CM START POINT ?
S26
NO
DISPLAY PICTURE
S27
YES
STOP DECODING AND DISPLAY
S28
1

FIG.110

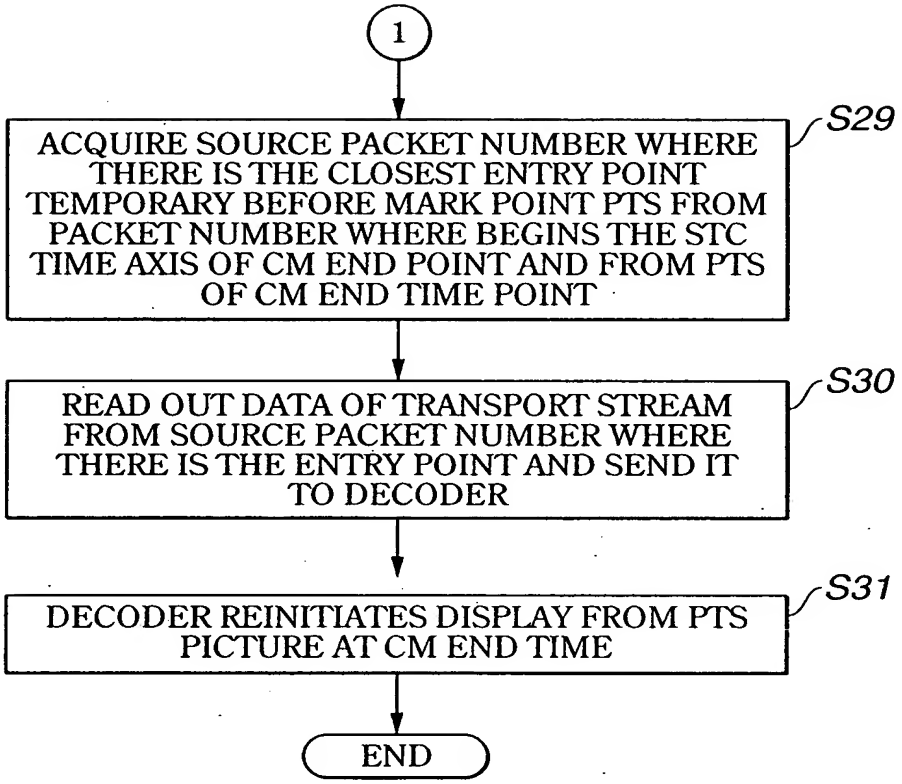

1
ACQUIRE SOURCE PACKET NUMBER WHERE THERE IS THE CLOSEST ENTRY POINT TEMPORARY BEFORE MARK POINT PTS FROM PACKET NUMBER WHERE BEGINS THE STC TIME AXIS OF CM END POINT AND FROM PTS OF CM END TIME POINT
S29
READ OUT DATA OF TRANSPORT STREAM FROM SOURCE PACKET NUMBER WHERE THERE IS THE ENTRY POINT AND SEND IT TO DECODER
S30
DECODER REINITIATES DISPLAY FROM PTS PICTURE AT CM END TIME
S31
END

FIG.111

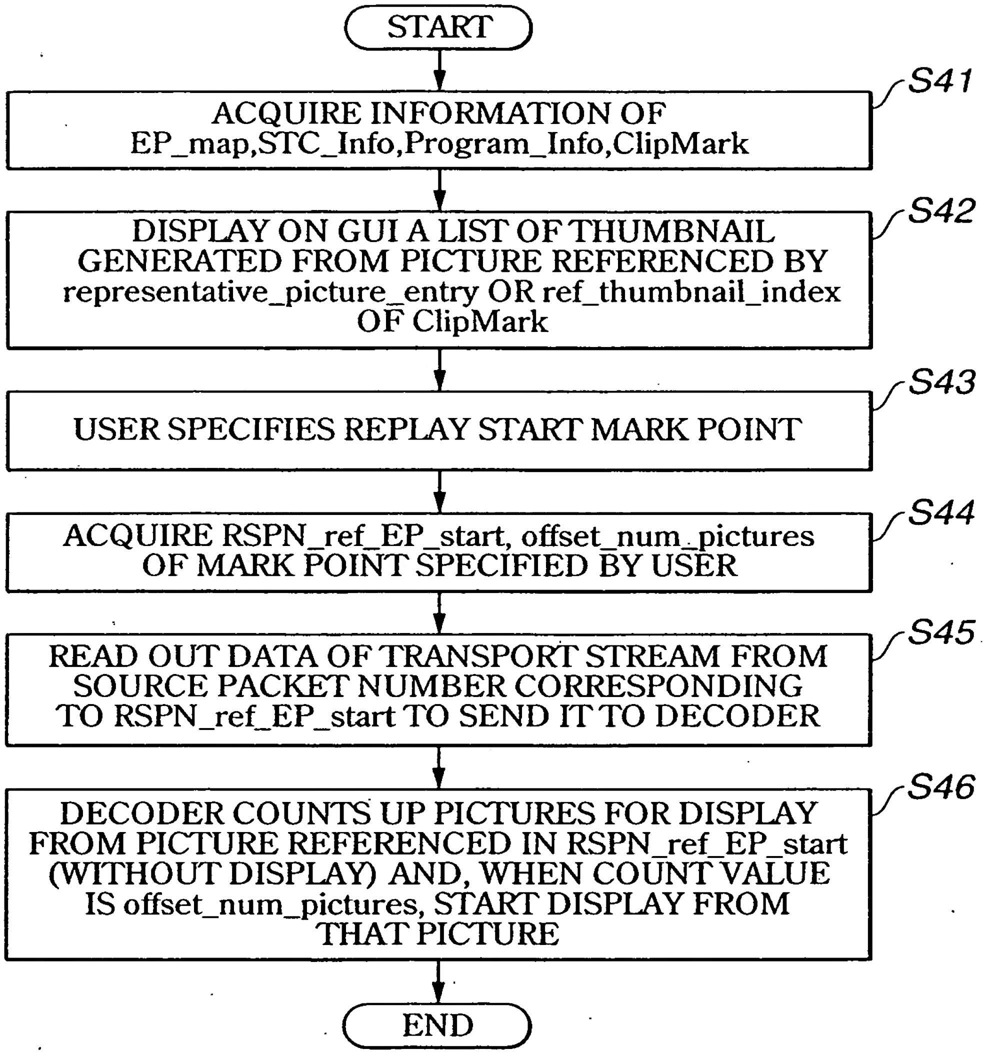

START
ACQUIRE INFORMATION OF EP_map,STC_Info,Program_Info,ClipMark
S41
DISPLAY ON GUI A LIST OF THUMBNAIL GENERATED FROM PICTURE REFERENCED BY representative_picture_entry OR ref_thumbnail_index OF ClipMark
S42
USER SPECIFIES REPLAY START MARK POINT
S43
ACQUIRE RSPN_ref_EP_start, offset_num_pictures OF MARK POINT SPECIFIED BY USER
S44
READ OUT DATA OF TRANSPORT STREAM FROM SOURCE PACKET NUMBER CORRESPONDING TO RSPN_ref_EP_start TO SEND IT TO DECODER
S45
DECODER COUNTS UP PICTURES FOR DISPLAY FROM PICTURE REFERENCED IN RSPN_ref_EP_start (WITHOUT DISPLAY) AND, WHEN COUNT VALUE IS offset_num_pictures, START DISPLAY FROM THAT PICTURE
S46
END

FIG.112

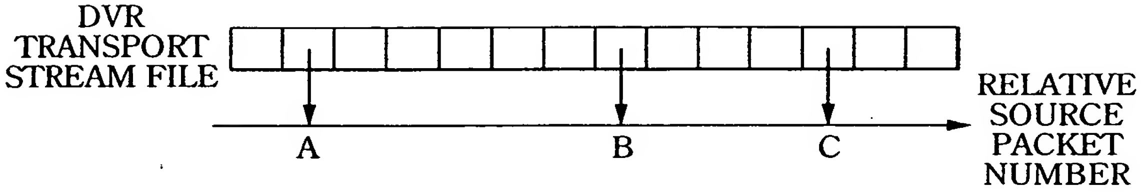


FIG.113

EP_map

| RSPN_EP_ start | PTS_EP_ start |
|---|---|
| . . . | . . . |
| A | PTS(A) |
| B | PTS(B) |
| C | PTS(C) |
| . . . | . . . |

FIG.114

ClipMark

| mark_type | mark_entry | | representative _ picture _entry | |
|---|---|---|---|---|
| | RSPN_ref_EP_ start | offset_num_ pictures | RSPN_ref_EP_ start | offset_num_ pictures |
| . . . | . . . | . . . | . . . | . . . |
| 0x92(scene start) | A | M1 | A | M2 |
| 0x94(CM start) | B | N1 | B | N1 |
| 0x95(CM end) | C | N2 | C | N2 |
| . . . | . . . | . . . | . . . | . . . |

FIG.115

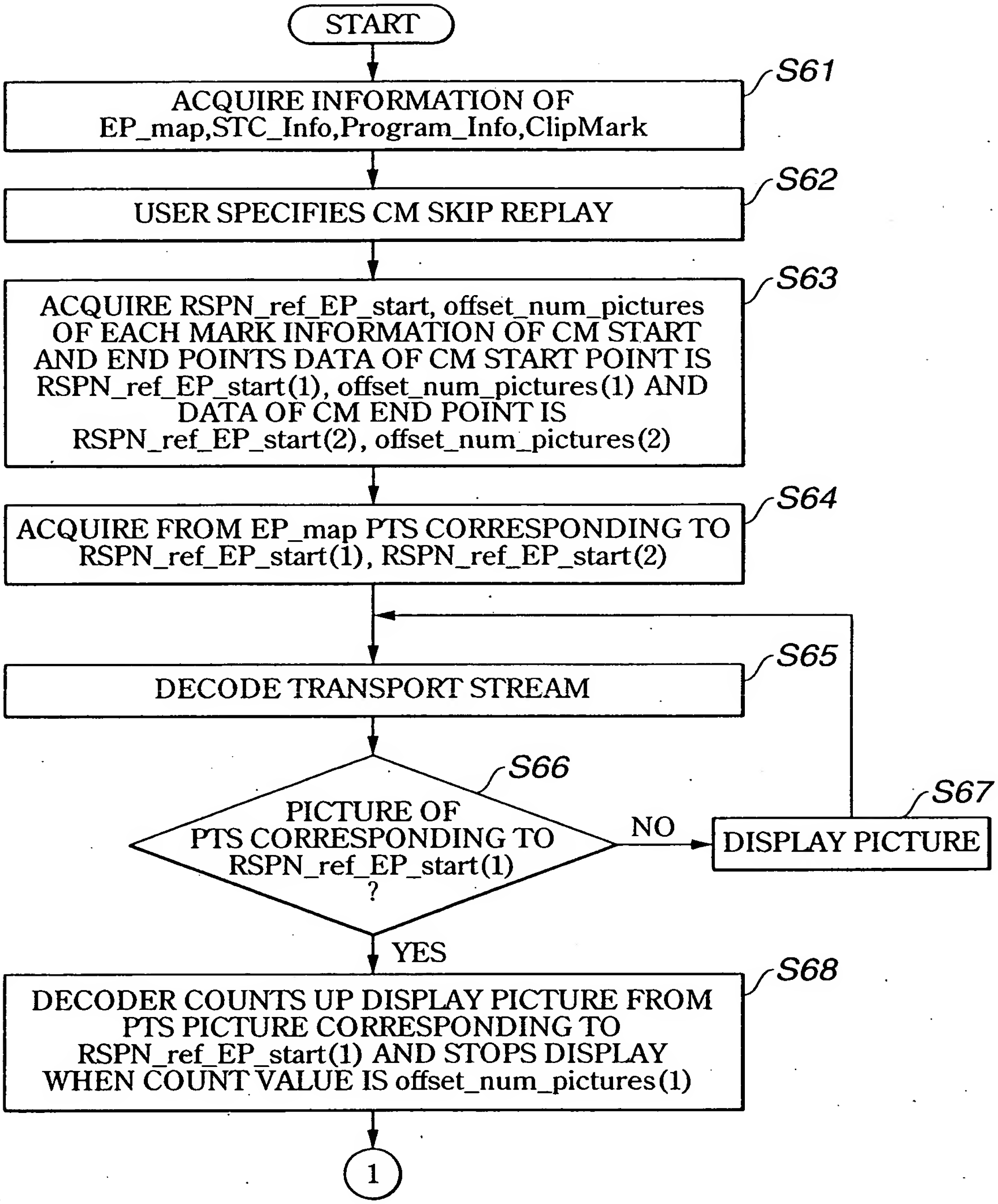

START
S61
ACQUIRE INFORMATION OF EP_map,STC_Info,Program_Info,ClipMark
S62
USER SPECIFIES CM SKIP REPLAY
S63
ACQUIRE RSPN_ref_EP_start, offset_num_pictures OF EACH MARK INFORMATION OF CM START AND END POINTS DATA OF CM START POINT IS RSPN_ref_EP_start(1), offset_num_pictures(1) AND DATA OF CM END POINT IS RSPN_ref_EP_start(2), offset_num_pictures(2)
S64
ACQUIRE FROM EP_map PTS CORRESPONDING TO RSPN_ref_EP_start(1), RSPN_ref_EP_start(2)
S65
DECODE TRANSPORT STREAM
S66
PICTURE OF PTS CORRESPONDING TO RSPN_ref_EP_start(1) ?
NO
S67
DISPLAY PICTURE
YES
S68
DECODER COUNTS UP DISPLAY PICTURE FROM PTS PICTURE CORRESPONDING TO RSPN_ref_EP_start(1) AND STOPS DISPLAY WHEN COUNT VALUE IS offset_num_pictures(1)
1

FIG.116

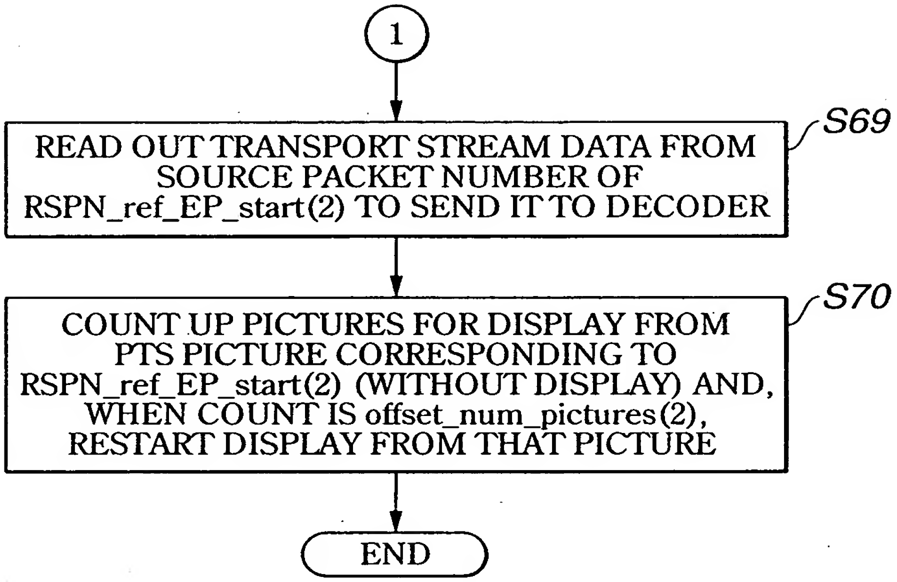

1
READ OUT TRANSPORT STREAM DATA FROM SOURCE PACKET NUMBER OF RSPN_ref_EP_start(2) TO SEND IT TO DECODER
S69
COUNT UP PICTURES FOR DISPLAY FROM PTS PICTURE CORRESPONDING TO RSPN_ref_EP_start(2) (WITHOUT DISPLAY) AND, WHEN COUNT IS offset_num_pictures(2), RESTART DISPLAY FROM THAT PICTURE
S70
END

FIG.117

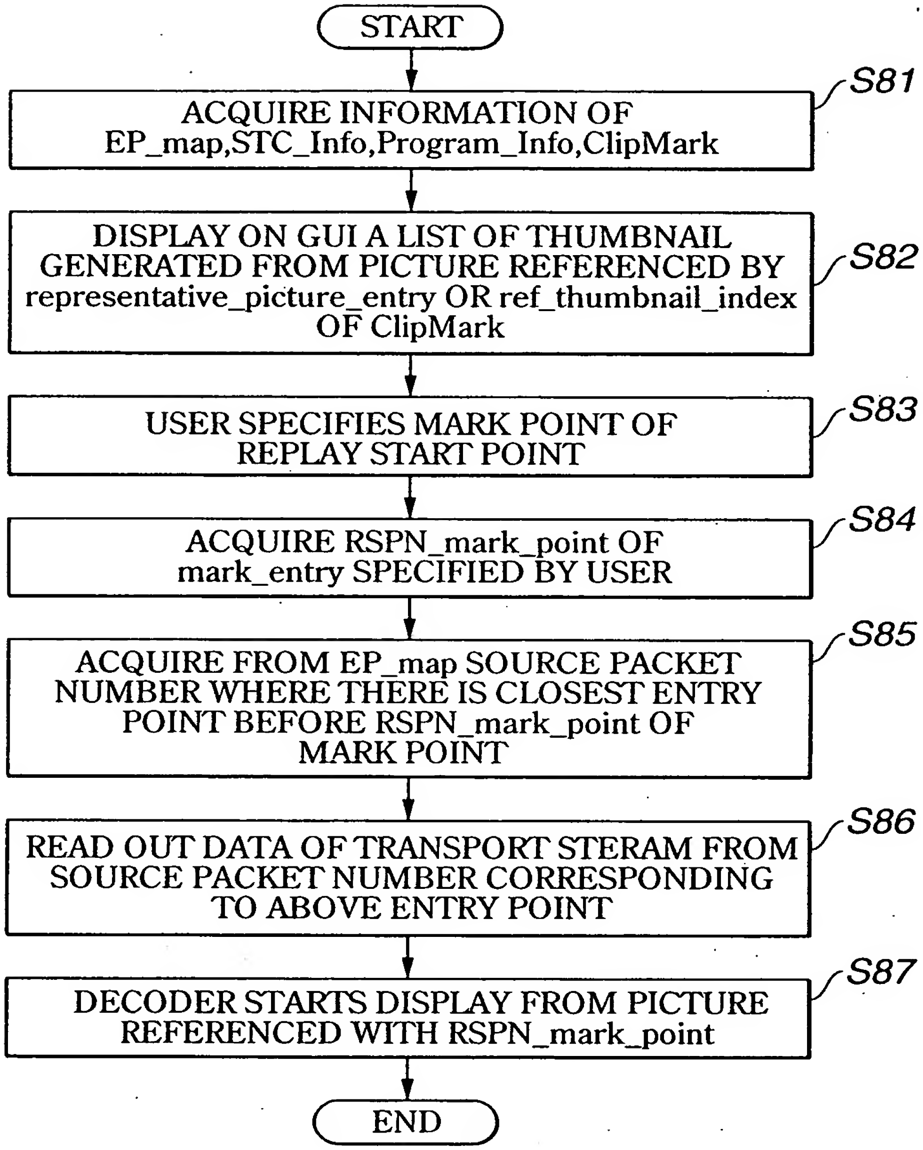

START
S81
ACQUIRE INFORMATION OF EP_map,STC_Info,Program_Info,ClipMark
S82
DISPLAY ON GUI A LIST OF THUMBNAIL GENERATED FROM PICTURE REFERENCED BY representative_picture_entry OR ref_thumbnail_index OF ClipMark
S83
USER SPECIFIES MARK POINT OF REPLAY START POINT
S84
ACQUIRE RSPN_mark_point OF mark_entry SPECIFIED BY USER
S85
ACQUIRE FROM EP_map SOURCE PACKET NUMBER WHERE THERE IS CLOSEST ENTRY POINT BEFORE RSPN_mark_point OF MARK POINT
S86
READ OUT DATA OF TRANSPORT STERAM FROM SOURCE PACKET NUMBER CORRESPONDING TO ABOVE ENTRY POINT
S87
DECODER STARTS DISPLAY FROM PICTURE REFERENCED WITH RSPN_mark_point
END

FIG.118

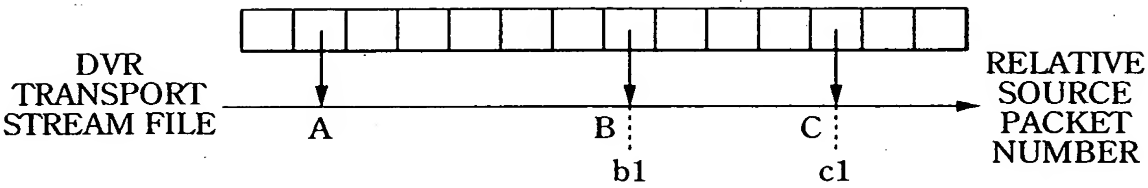


FIG.119

EP_map

| RSPN_EP_start | PTS_EP_start |
|---|---|
| . . . | . . . |
| A | PTS(A) |
| B | PTS(B) |
| C | PTS(C) |
| . . . | . . . |

FIG.120

ClipMark

| mark_type | mark_entry | representative _ picture _entry |
|---|---|---|
| | RSPN_mark_point | RSPN_mark_point |
| . . . | . . . | . . . |
| 0x92(scene start) | a1 | a2 |
| 0x94(CM start) | b1 | b1 |
| 0x95(CM end) | c1 | c1 |
| . . . | . . . | . . . |

FIG.121

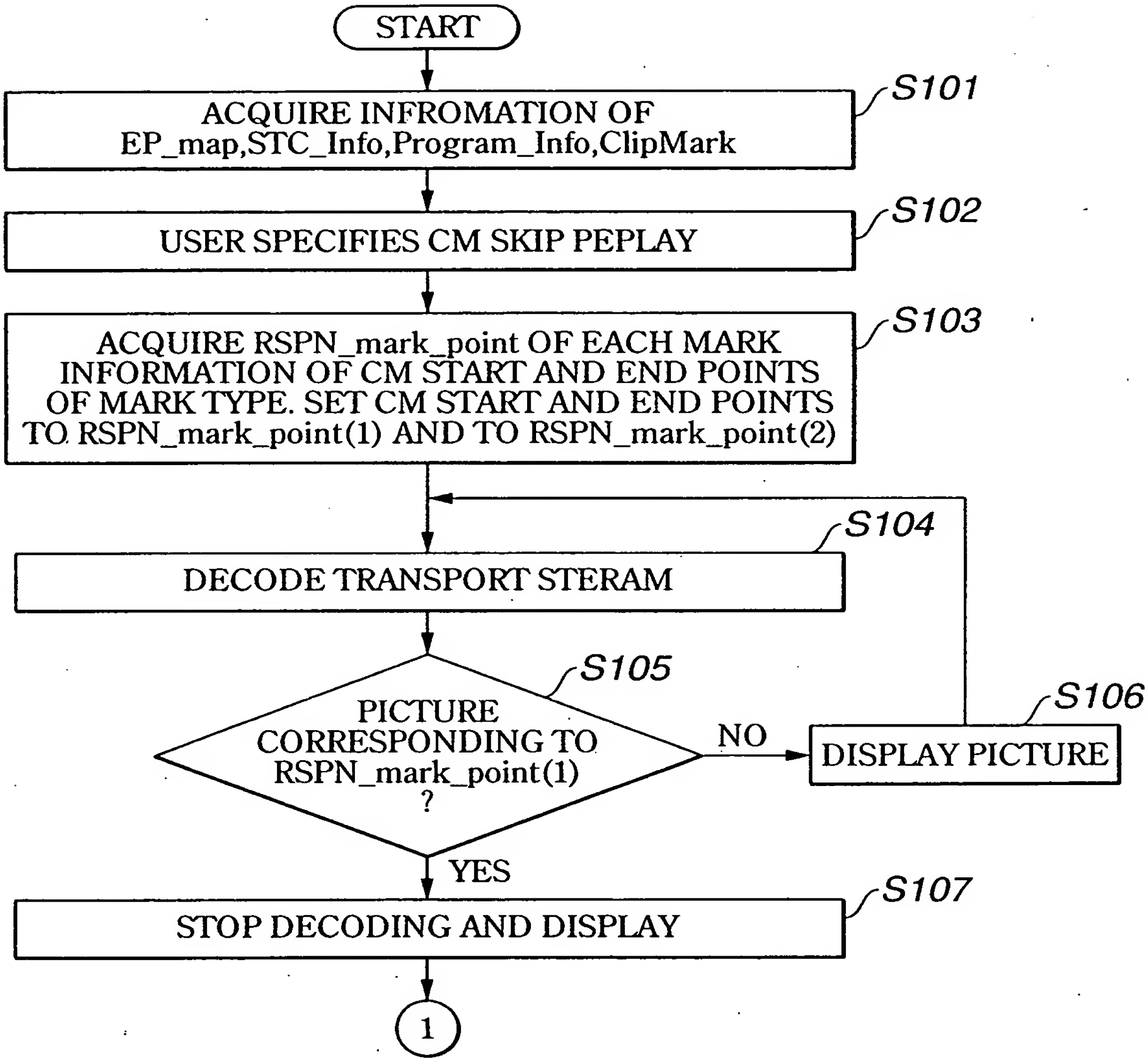

START
ACQUIRE INFROMATION OF EP_map,STC_Info,Program_Info,ClipMark
S101
USER SPECIFIES CM SKIP PEPLAY
S102
ACQUIRE RSPN_mark_point OF EACH MARK INFORMATION OF CM START AND END POINTS OF MARK TYPE. SET CM START AND END POINTS TO RSPN_mark_point(1) AND TO RSPN_mark_point(2)
S103
DECODE TRANSPORT STERAM
S104
PICTURE CORRESPONDING TO RSPN_mark_point(1) ?
S105
NO
DISPLAY PICTURE
S106
YES
STOP DECODING AND DISPLAY
S107
1

FIG.122

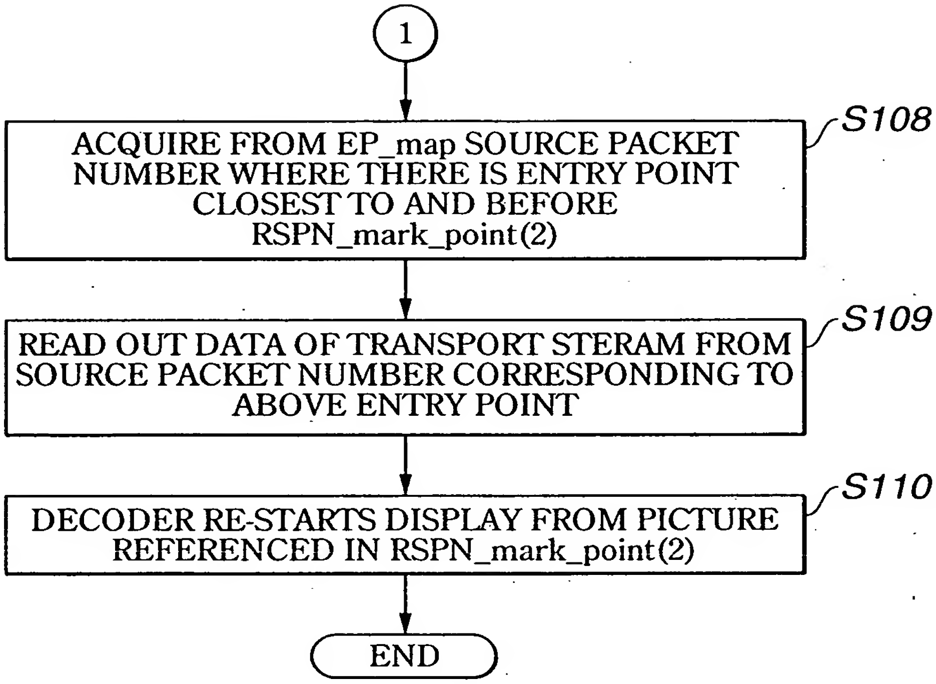

1
ACQUIRE FROM EP_map SOURCE PACKET NUMBER WHERE THERE IS ENTRY POINT CLOSEST TO AND BEFORE RSPN_mark_point(2)
S108
READ OUT DATA OF TRANSPORT STERAM FROM SOURCE PACKET NUMBER CORRESPONDING TO ABOVE ENTRY POINT
S109
DECODER RE-STARTS DISPLAY FROM PICTURE REFERENCED IN RSPN_mark_point(2)
S110
END

FIG.123

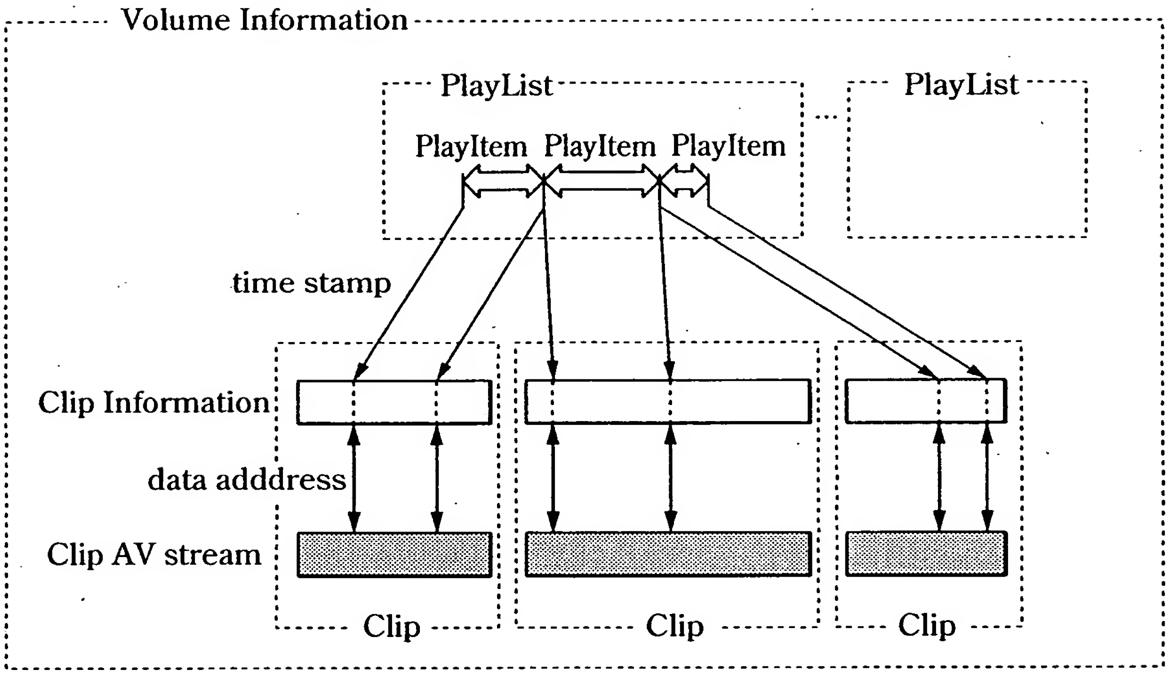


FIG.124

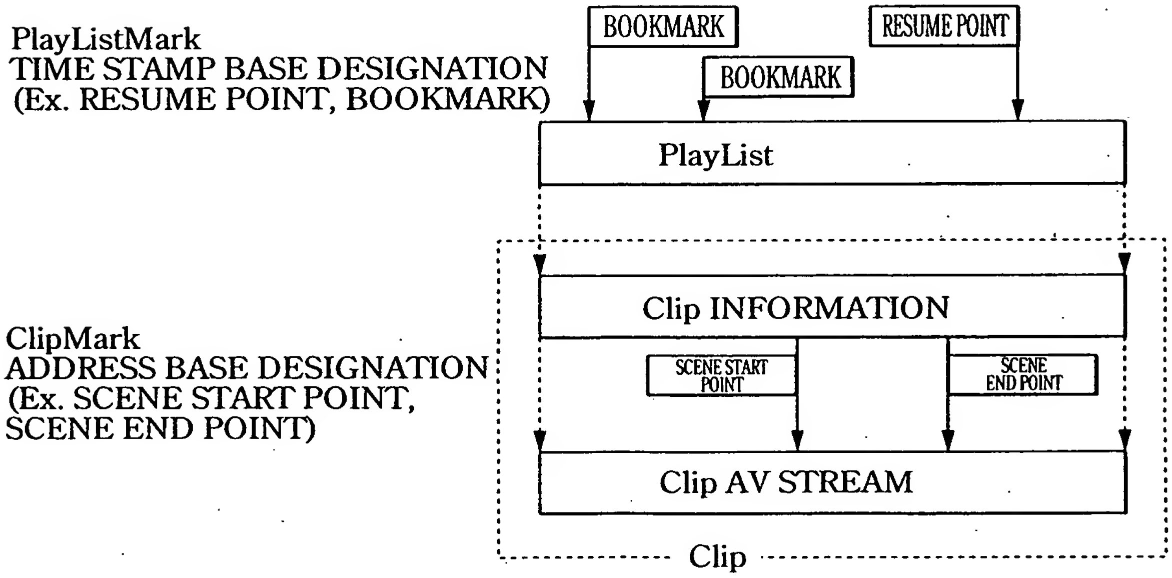


FIG.125

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| ClipMark() { | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| **number_of_Clip_marks** | 16 | uimsbf |
| for (i=0; i<*number_of_Clip_marks*; i++) { | | |
| reserved | 8 | bslbf |
| **mark_type** | 8 | bslbf |
| **RSPN_mark** | 32 | uimsbf |
| reserved | 32 | bslbf |
| **ref_thumbnail_index** | 16 | uimsbf |
| } | | |
| } | | |

# FIG.126

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| ClipMark() { | | |
| **version_number** | 8*4 | bslbf |
| **length** | 32 | uimsbf |
| **number_of_Clip_marks** | 16 | uimsbf |
| for (i=0; i<*number_of_Clip_marks*; i++) { | | |
| reserved | 8 | bslbf |
| **mark_type** | 8 | bslbf |
| **RSPN_ref_EP_start** | 32 | uimsbf |
| **offset_num_pictures** | 32 | uimsbf |
| **ref_thumbnail_index** | 16 | uimsbf |
| } | | |
| } | | |

# FIG.127

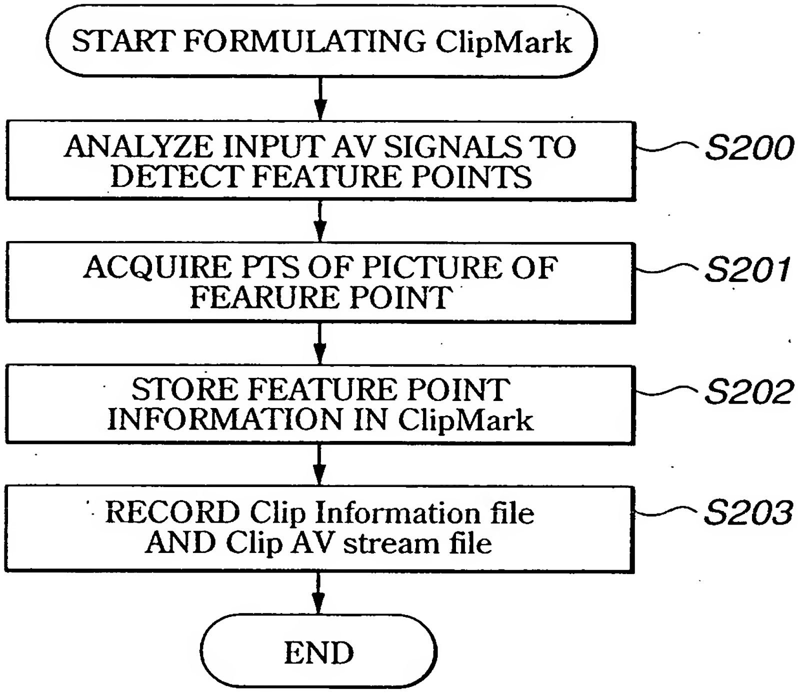

START FORMULATING ClipMark
ANALYZE INPUT AV SIGNALS TO DETECT FEATURE POINTS
S200
ACQUIRE PTS OF PICTURE OF FEARURE POINT
S201
STORE FEATURE POINT INFORMATION IN ClipMark
S202
RECORD Clip Information file AND Clip AV stream file
S203
END

FIG.128

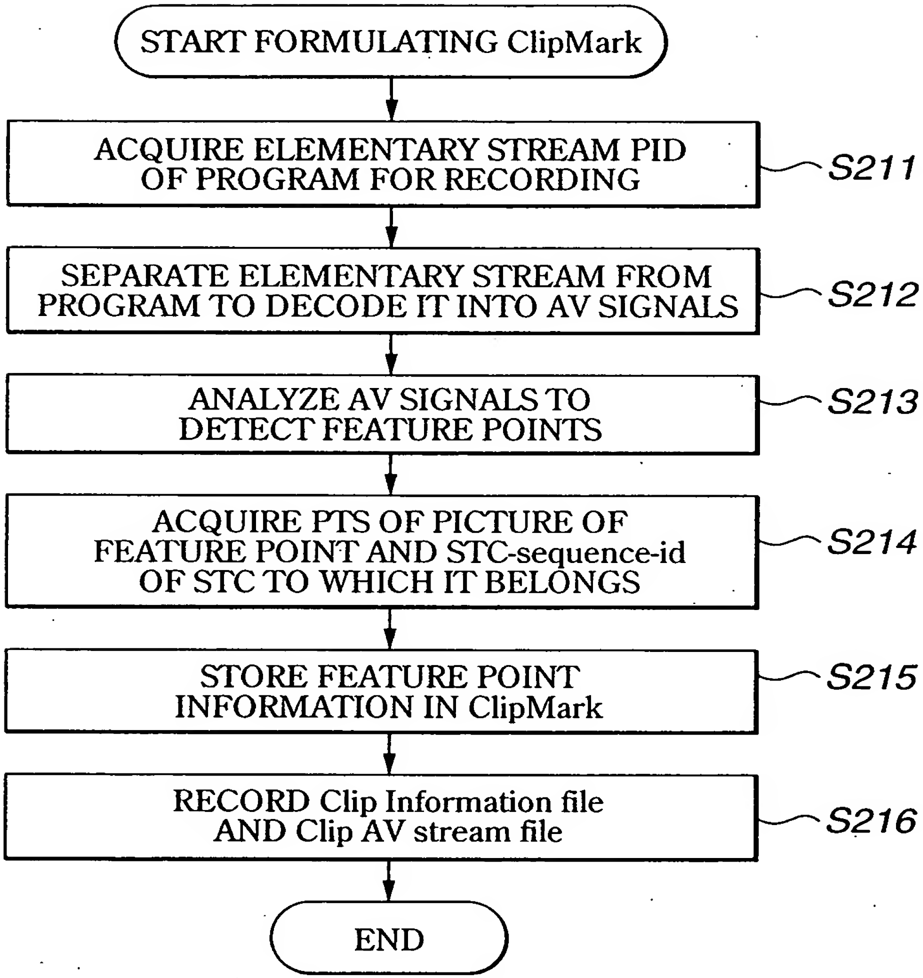

START FORMULATING ClipMark
ACQUIRE ELEMENTARY STREAM PID OF PROGRAM FOR RECORDING
S211
SEPARATE ELEMENTARY STREAM FROM PROGRAM TO DECODE IT INTO AV SIGNALS
S212
ANALYZE AV SIGNALS TO DETECT FEATURE POINTS
S213
ACQUIRE PTS OF PICTURE OF FEATURE POINT AND STC-sequence-id OF STC TO WHICH IT BELONGS
S214
STORE FEATURE POINT INFORMATION IN ClipMark
S215
RECORD Clip Information file AND Clip AV stream file
S216
END

FIG.129

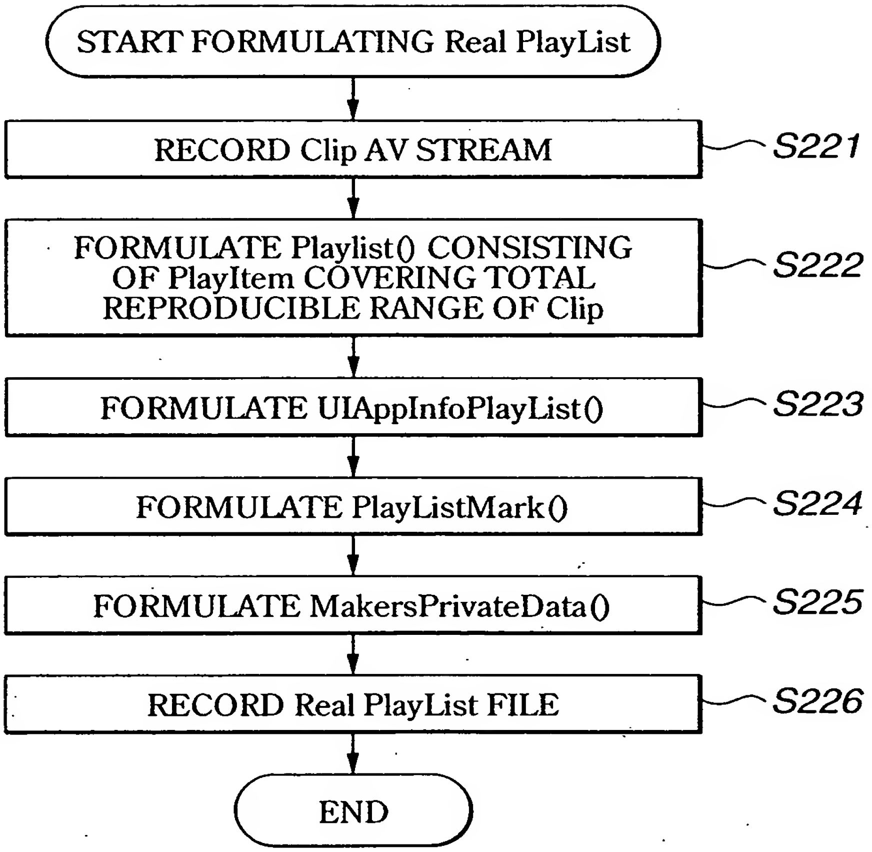

START FORMULATING Real PlayList
RECORD Clip AV STREAM
S221
FORMULATE Playlist() CONSISTING OF PlayItem COVERING TOTAL REPRODUCIBLE RANGE OF Clip
S222
FORMULATE UIAppInfoPlayList()
S223
FORMULATE PlayListMark()
S224
FORMULATE MakersPrivateData()
S225
RECORD Real PlayList FILE
S226
END

FIG.130

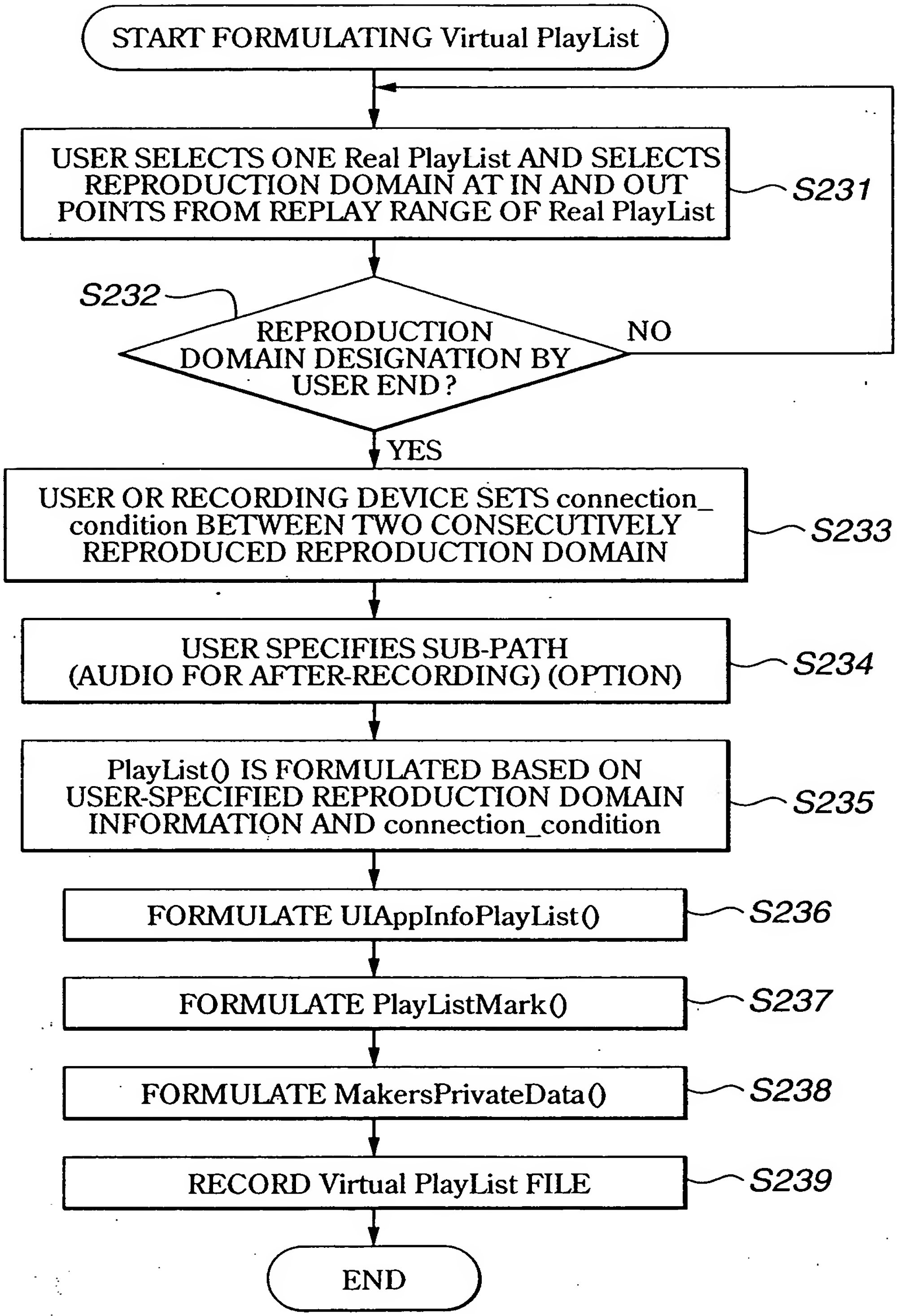

START FORMULATING Virtual PlayList
USER SELECTS ONE Real PlayList AND SELECTS REPRODUCTION DOMAIN AT IN AND OUT POINTS FROM REPLAY RANGE OF Real PlayList
S231
S232
REPRODUCTION DOMAIN DESIGNATION BY USER END ?
NO
YES
USER OR RECORDING DEVICE SETS connection_condition BETWEEN TWO CONSECUTIVELY REPRODUCED REPRODUCTION DOMAIN
S233
USER SPECIFIES SUB-PATH (AUDIO FOR AFTER-RECORDING) (OPTION)
S234
PlayList() IS FORMULATED BASED ON USER-SPECIFIED REPRODUCTION DOMAIN INFORMATION AND connection_condition
S235
FORMULATE UIAppInfoPlayList()
S236
FORMULATE PlayListMark()
S237
FORMULATE MakersPrivateData()
S238
RECORD Virtual PlayList FILE
S239
END

FIG.131

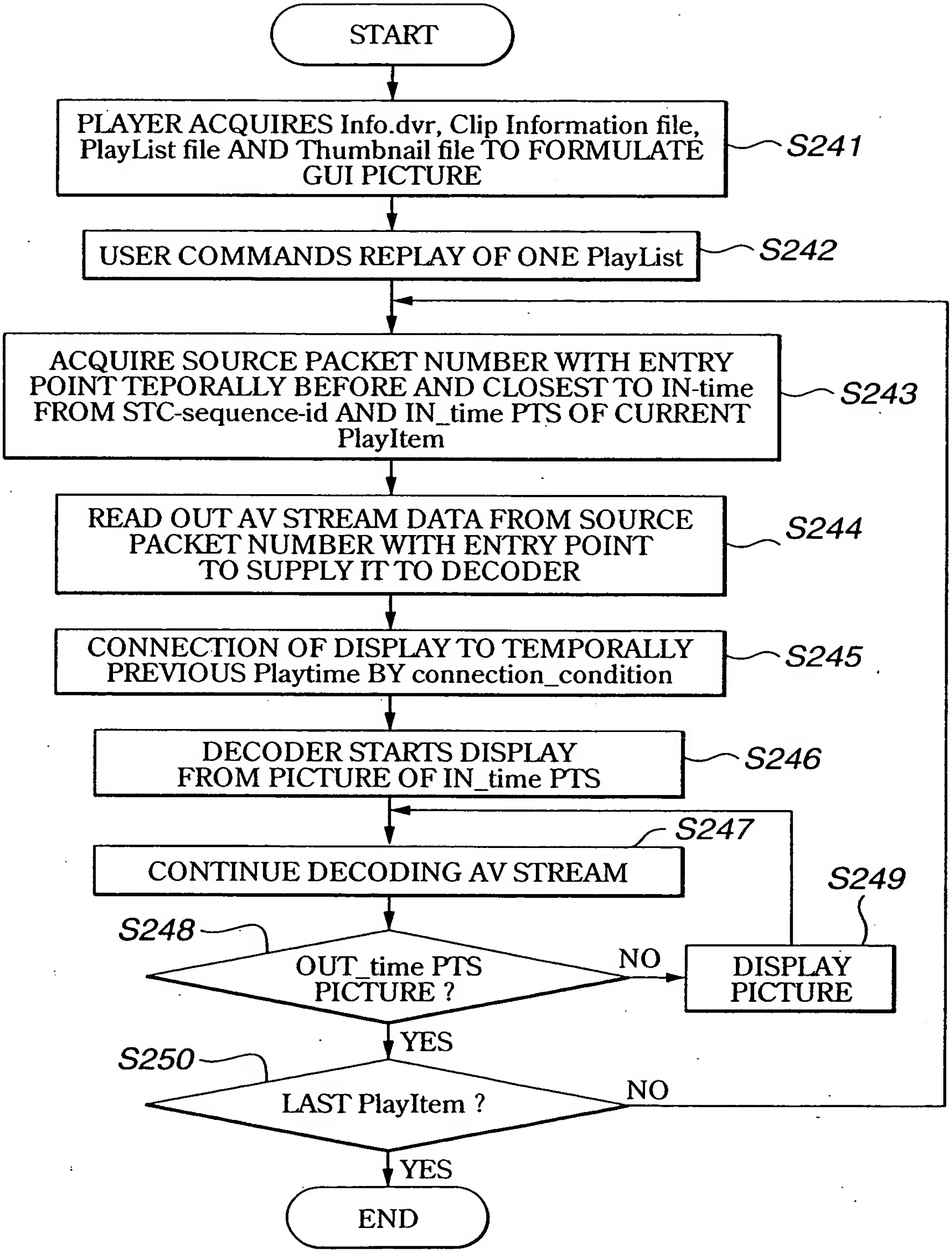

START
PLAYER ACQUIRES Info.dvr, Clip Information file, PlayList file AND Thumbnail file TO FORMULATE GUI PICTURE
S241
USER COMMANDS REPLAY OF ONE PlayList
S242
ACQUIRE SOURCE PACKET NUMBER WITH ENTRY POINT TEPORALLY BEFORE AND CLOSEST TO IN-time FROM STC-sequence-id AND IN_time PTS OF CURRENT PlayItem
S243
READ OUT AV STREAM DATA FROM SOURCE PACKET NUMBER WITH ENTRY POINT TO SUPPLY IT TO DECODER
S244
CONNECTION OF DISPLAY TO TEMPORALLY PREVIOUS Playtime BY connection_condition
S245
DECODER STARTS DISPLAY FROM PICTURE OF IN_time PTS
S246
S247
CONTINUE DECODING AV STREAM
S249
S248
OUT_time PTS PICTURE ?
NO
DISPLAY PICTURE
YES
S250
LAST PlayItem ?
NO
YES
END

FIG.132

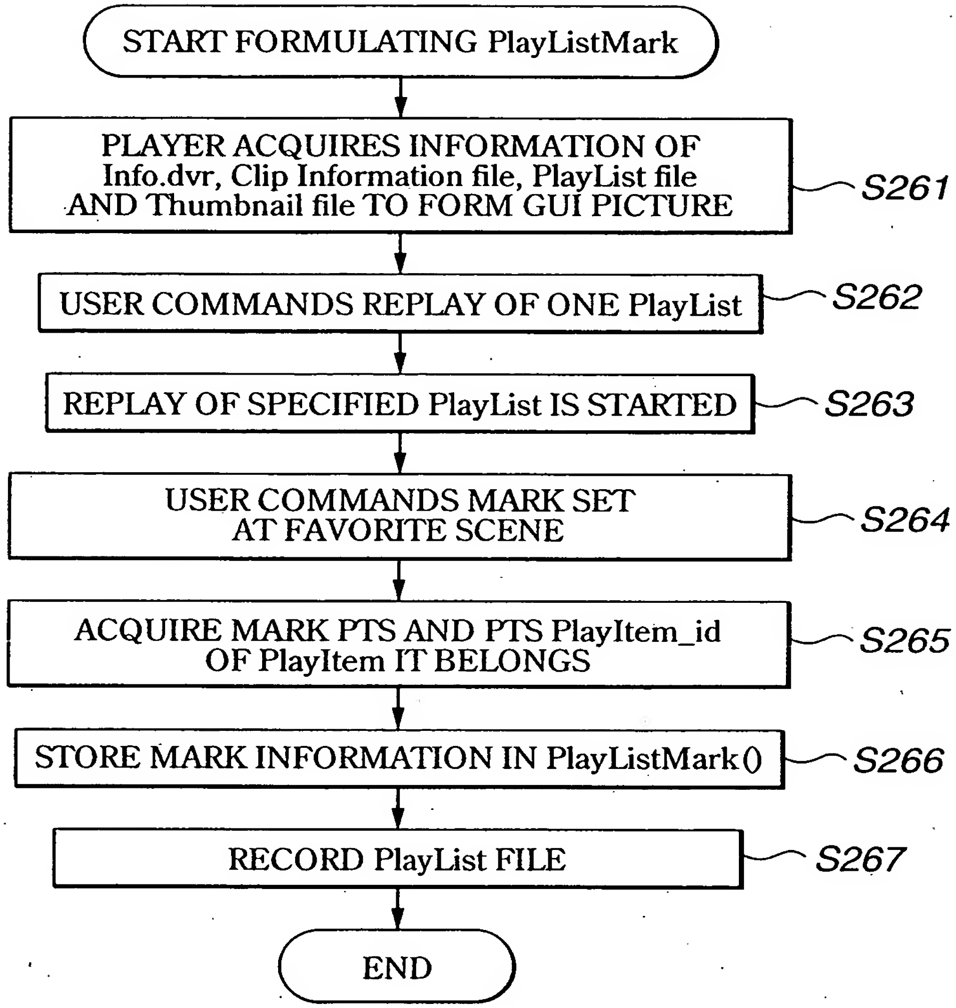

START FORMULATING PlayListMark
PLAYER ACQUIRES INFORMATION OF Info.dvr, Clip Information file, PlayList file AND Thumbnail file TO FORM GUI PICTURE
S261
USER COMMANDS REPLAY OF ONE PlayList
S262
REPLAY OF SPECIFIED PlayList IS STARTED
S263
USER COMMANDS MARK SET AT FAVORITE SCENE
S264
ACQUIRE MARK PTS AND PTS PlayItem_id OF PlayItem IT BELONGS
S265
STORE MARK INFORMATION IN PlayListMark()
S266
RECORD PlayList FILE
S267
END

FIG.133

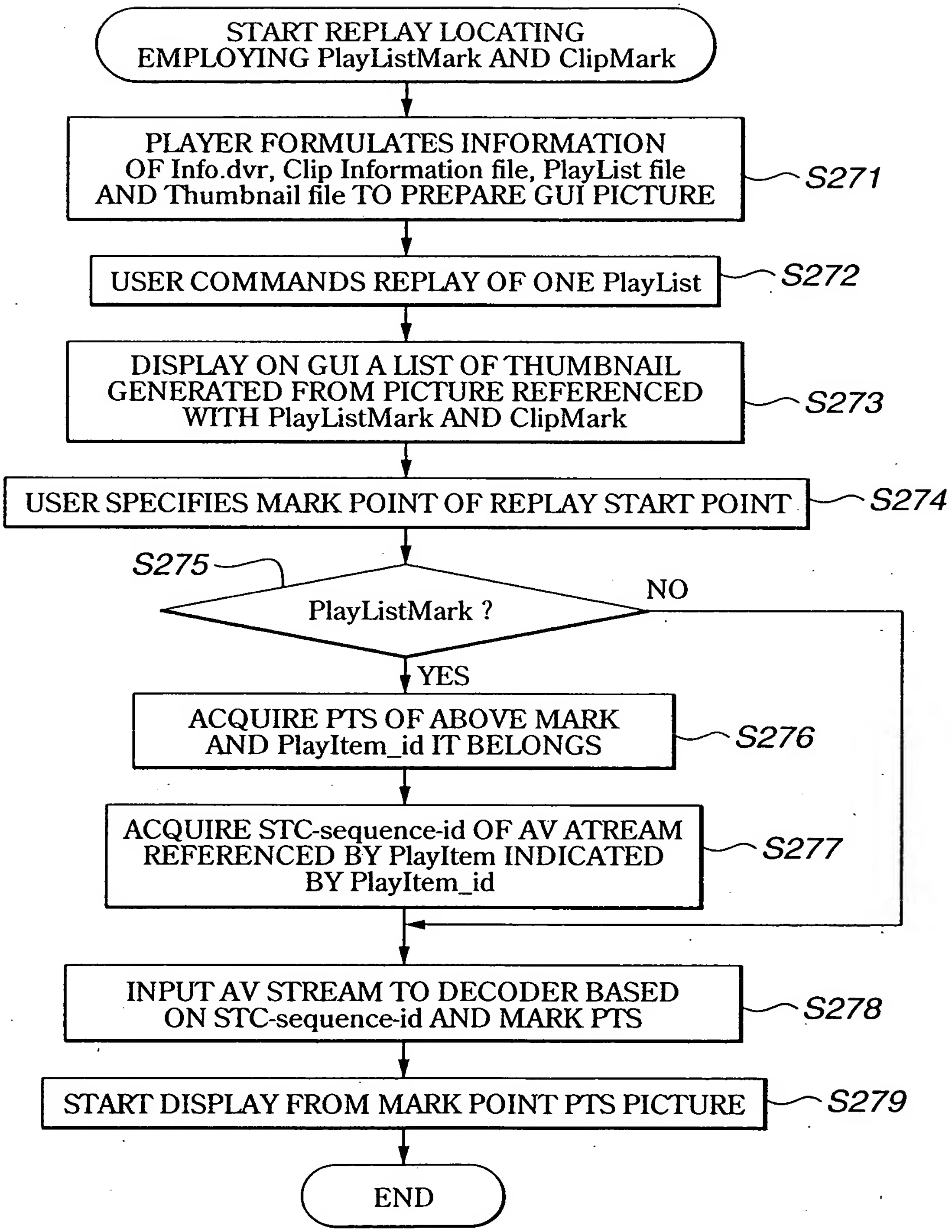

START REPLAY LOCATING EMPLOYING PlayListMark AND ClipMark
PLAYER FORMULATES INFORMATION OF Info.dvr, Clip Information file, PlayList file AND Thumbnail file TO PREPARE GUI PICTURE
S271
USER COMMANDS REPLAY OF ONE PlayList
S272
DISPLAY ON GUI A LIST OF THUMBNAIL GENERATED FROM PICTURE REFERENCED WITH PlayListMark AND ClipMark
S273
USER SPECIFIES MARK POINT OF REPLAY START POINT
S274
S275
PlayListMark ?
NO
YES
ACQUIRE PTS OF ABOVE MARK AND PlayItem_id IT BELONGS
S276
ACQUIRE STC-sequence-id OF AV ATREAM REFERENCED BY PlayItem INDICATED BY PlayItem_id
S277
INPUT AV STREAM TO DECODER BASED ON STC-sequence-id AND MARK PTS
S278
START DISPLAY FROM MARK POINT PTS PICTURE
S279
END

FIG.134

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| PlayListMark() { | | |
| **length** | 32 | uimsbf |
| **number_of_PlayList_marks** | 16 | uimsbf |
| for (i=0;i<*number_of_PlayList_marks*;i++) { | | |
| **mark_invalid_flag** | 1 | uimsbf |
| **mark_type** | 7 | uimsbf |
| **mark_name_length** | 8 | uimsbf |
| **ref_to_PlayItem_id** | 16 | uimsbf |
| **mark_time_stamp** | 32 | uimsbf |
| **entry_ES_PID** | 16 | uimsbf |
| **ref_to_thumbnail_index** | 16 | uimsbf |
| **mark_name** | 8*32 | bslbf |
| } | | |
| } | | |

FIG.135

| VALUES | MEANING | NOTS |
|---|---|---|
| 0x00 | Resume-mark | REPLAY RESUME POINT. THE NUMBER OF REPLAY RESUME POINTS DEFINED IN PlayListMark() MUST BE 0 OR 1. |
| 0x01 | Book-mark | REPLAY ENTRY POINT OF PlayList. THIS MARK CAN BE SET BY USER AND USED AS MARK FOR SPECIFYING START POINT OF FAVORITE SCENE. A PLURAL NUMBER OF THIS MARK MAY BE PROVIDED IN PlayListMark(). |
| 0x02 | Chapter-mark | USER INTENDS THAT A CHAPTER IN PlayList BEGINS WITH THIS MARK. A PLURAL NUMBER OF THIS MARK MAY BE PROVIDED IN PlayListMARK(). |
| 0x03 | Skip-start-mark | IF ONE Skip-start-mark IS SET IN PlayListMark, ONE Skip-end-mark MUST BE SET DIRECTLY AFTER THE Skip-start-mark ENTRY. USER INTENDS TO SKIP PlayList REPLAY FROM TIME STAMP OF Skip-start-mark TO TIME STAMP OF Skip-end-mark. Skip-start-mark HAS SAME MEANING AS Skip-end-mark. IF entry-ES-PID IS NOT 0xFFFF, Skip-start-mark HAS SAME entry-ES-PID VALUE AS Skip-end-mark. THIS MARK CAN BE SET BY USER. THERE MAY BE PLURAL SUCH MARKS IN PlayListMark(). |
| 0x04 | Skip-end-mark | |
| 0x05-0x3F | Reserved for future use | Reserved for PlayListMark |
| 0x40-0x7F | Reserved for ClipMark | |

FIG.136

| SYNTAX | NUMBER OF BYTES | ABBREVIATION |
|---|---|---|
| ClipMark() { | | |
| **length** | 32 | uimsbf |
| **maker_ID** | 16 | uimsbf |
| **number_of_Clip_marks** | 16 | uimsbf |
| for (i=0; i<*number_of_Clip_marks*; i++) { | | |
| **mark_invalid_flag** | 1 | uimsbf |
| **mark_type** | 7 | uimsbf |
| **ref_to_STC_id** | 8 | uimsbf |
| **mark_time_stamp** | 32 | uimsbf |
| **entry_ES_PID** | 16 | uimsbf |
| **ref_to_thumbnail_index** | 16 | uimsbf |
| **representative_picture_time_stamp** | 32 | uimsbf |
| } | | |
| } | | |

FIG.137

| Mark_type | MEANING | NOTES |
|---|---|---|
| 0x00-<br>0x3F | reserved for future use | Reserved for PlayListMark |
| 0x40 | Scene-start-mark | MARK POINT INDICATING SCENE START POINT |
| 0x41-<br>0x5F | Reserved for common ClipMark | |
| 0x60-<br>0x7F | Maker defined ClipMark | MAKER SPECIFIED WITH maker_ID CAN FREELY DEFINE MEANING |

FIG.138

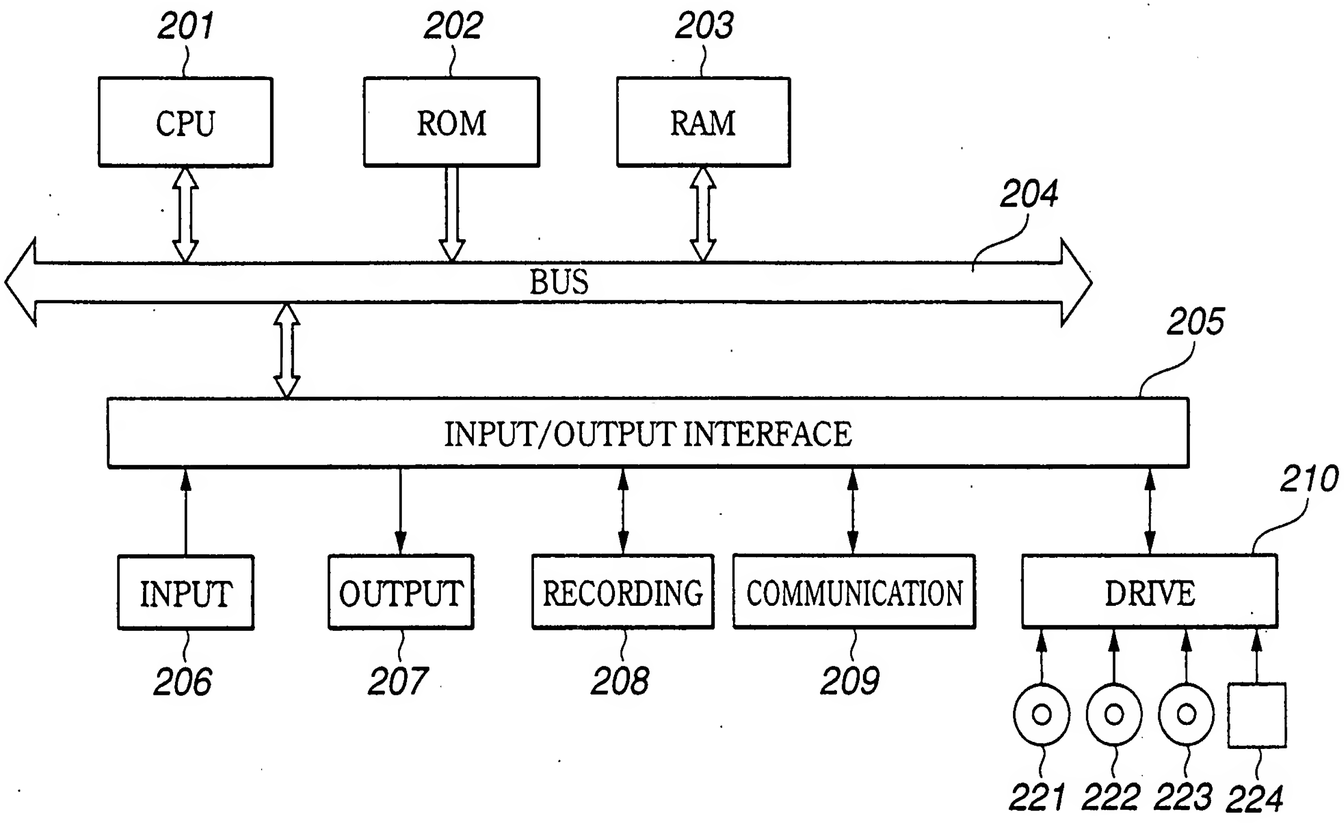

201
CPU
202
ROM
203
RAM
204
BUS
205
INPUT/OUTPUT INTERFACE
210
INPUT
206
OUTPUT
207
RECORDING
208
COMMUNICATION
209
DRIVE
221
222
223
224

FIG.139